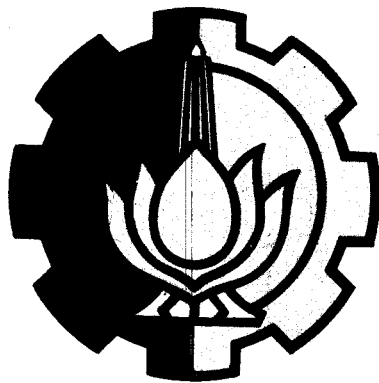
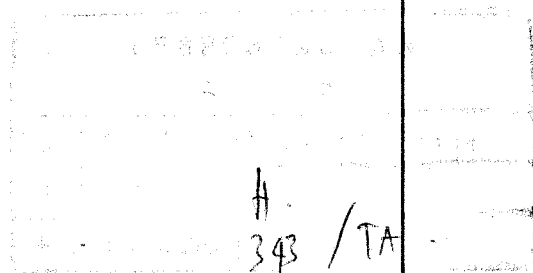


389A / II / 4 / 91 ✓

INTERFACE PAPAN GRAFIK PADA KOMPUTER IBM PC



PSE
621.398
Fit
I-1
1990

Disusun oleh :

ARMY FITHRY

2842200118

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
S U R A B A Y A
1990

INTERFACE PAPAN GRAFIK PADA KOMPUTER IBM PC

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

P a d a

**Bidang Studi Teknik Komputer
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a**

**Mengetahui / Menyetujui
Dosen Pembimbing**



Ir. SOEPENO Dj. M.Sc. Ph. D.



Ir. S. HARDIRISTANTO

**SURABAYA
NOPEMBER 1990**

ABSTRAK

Dewasa ini aplikasi komputer semakin meluas ke berbagai bidang, terutama setelah beredarnya komputer mikro.

Sejalan dengan itu, komputer pun mulai digunakan sebagai alat bantu gambar.

Untuk itu dibutuhkan suatu perangkat lunak yang dapat memberikan fasilitas grafik.

Dengan peralatan input grafik yang interaktif, maka pembuatan gambar pun menjadi semakin mudah.

KATA PENGANTAR

Syukur Alhamdulillah kami panjatkan ke hadapan Allah SWT yang telah memberikan hidayah dan inayahNya sehingga kami dapat menyelesaikan tugas akhir yang berjudul "INTERFACE PAPAN GRAFIK PADA KOMPUTER IBM PC" dengan sebaik-baiknya.

Tugas akhir ini disusun guna memenuhi salah satu persyaratan untuk menyelesaikan studi di bidang studi Teknik Komputer, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Banyak sekali bantuan yang sangat berarti yang datang dari berbagai pihak. Untuk itu sudah selayaknyalah bila kami ingin mengucapkan terima kasih kepada :

1. Bapak Ir. Syariffuddin Mahmudsyah M. Eng selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri ITS.
2. Bapak Ir. S. Hardirianto selaku dosen

- pembimbing yang banyak memberikan bimbingan pada kami dalam menyelesaikan Tugas Akhir ini.
3. Bapak Ir. Soepeno Djanali M.Sc. Ph.D. selaku Ketua Bidang Studi Teknik Komputer Jurusan Teknik Elektro Fakultas Teknologi Industri ITS yang sekaligus sebagai dosen pembimbing dan dosen wali kami.
 4. Ibu dan saudara-saudara kami yang tidak henti-hentinya memberikan semangat kepada kami.
 5. Rekan-rekan warga lab. B-201, antara lain : Kartolo, Linggo, Spock, Raden, Tralala, Sodron, Oblik, Basuki dan lain-lain.
 6. Rekan-rekan lain yang tidak mungkin kami sebutkan satu persatu di sini.

Kami sadar bahwa Tugas Akhir kami ini masih jauh dari sempurna, untuk itu saran dan kritik dari berbagai pihak tentulah akan sangat bermanfaat bagi kami untuk masa-masa yang akan datang.

Akhirul kata, kami hanya bisa berharap semoga buku Tugas Akhir ini dapat memberi manfaat bagi yang membacanya.

Surabaya, Nopember 1990

Penyusun

DAFTAR ISI

Bab	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Permasalahan dan Pembatasannya	2
1.4. Sistematika Pembahasan	5
II. TEORI PENUNJANG	6
2.1. Arsitektur Mikrokomputer IBM	7
2.1.1. Mikroprosesor 8088	9
2.1.2. Rangkaian Clock	11

2.1.3. Sistem Bus	12
2.1.4. Sistem ROM	12
2.1.5. Sistem RAM	13
2.1.6. Sistem Timer/Counter	14
2.1.7. Sistem DMA	14
2.1.8. Sistem Interrupt	15
2.1.9. Sistem Adapter	15
2.2. Slot Pada IBM PC-XT	16
2.3. Pengalamatan Port I/O	20
2.3.1. Pemetaan Alamat Port I/O	21
2.4. Pemetaan Memori	22
2.5. Tampilan Pada Komputer IBM PC	27
2.5.1. Adapter Tampilan	27
2.5.1.1. Memori dan Adapter Tampilan	27
2.5.1.2. Menampilkan ke Layar ..	28
2.5.2. Mode Tampilan Video	28
2.5.2.1. Resolusi Tampilan	29
2.5.3. Pemakaian Warna	30
2.5.3.1. Warna Pada Mode Teks ..	32
2.5.3.2. Warna Pada Mode Grafik	33
2.6. Dasar-Dasar Komputer Grafik	33
2.6.1. Sistem Koordinat	33
2.6.1.1. Koordinat Cartesien (Rectangular)	35
2.6.1.1.1. Kelinieran .	35
2.6.1.1.2. Arah dan Titik Asal ...	36

2.6.1.2. Koordinat Polar	37
2.6.2. Titik	39
2.6.3. Lokasi Titik	40
2.6.4. Garis	40
2.6.5. Kurva	41
2.6.6. Fill	41
2.6.7. Algoritma Garis	41
2.6.7.1. Raster	43
2.6.7.2. Algoritma Bresenham Un- tuk Garis	44
2.6.7.3. Pola Garis	44
2.6.7.4. Lebar Garis	44
2.6.7.5. Menghilangkan Patahan Garis (Antialiasing) ..	45
2.6.8. Algoritma Lingkaran	46
2.6.8.1. Algoritma Bresenham Un- tuk Lingkaran	46
2.6.8.2. Pola Lingkaran	47
2.6.8.3. Lebar Lingkaran	48
2.6.9. Busur	48
2.6.9.1. Lingkaran Dengan 3 Ti- tik	49
2.6.10. Algoritma Fill	51
2.6.10.1. Flood	51
2.6.10.2. Memuluskan Gambar (Smoothing)	52
2.6.10.3. Pengecatan (Brush- ing)	53

2.6.10.4. Palette	54
2.6.11. Penyilangan (Crosshatching) ...	54
2.6.12. Format File Gambar	55
2.6.13. Pemotongan Gambar (Clipping) ..	55
2.6.13.1. Clipping Window	56
2.7. Peralatan Input Bagi Komputer Grafik ..	56
2.7.1. Konversi Data Input.....	58
2.7.1.1. Metoda Analog to Digital Converter	58
2.7.1.2. Metoda Timer	59
2.8. Analog to Digital Converter	60
2.8.1. Flash ADC	61
2.8.2. Successive Approximation ADC ...	63
III. PERENCANAAN DAN PEMBUATAN PERANGKAT KERAS	66
3.1. Perangkat Keras Untuk IBM PC-XT	67
3.2. Rangkaian Output Untuk IBM PC-XT	69
3.2.1. Rangkaian Enable Output	70
3.2.2. Output Write Strobe	72
3.2.3. Board Write Strobe	72
3.2.4. Output Latch	73
3.3. Rangkaian Input Untuk IBM PC-XT	75
3.3.1. Rangkaian Enable Input	75
3.3.2. Output Read Strobe	75
3.3.3. Board Read Strobe	76
3.3.4. Input Buffer	77
3.4. Rangkaian ADC	79
IV. PERENCANAAN DAN PEMBUATAN PERANGKAT LUNAK	84
4.1. Struktur Program	84

4.2. Manipulasi Memori	86
4.3. Perangkat Keras dan Mode Tampilan	86
4.4. Perangkat Lunak	87
4.5. Fasilitas Bagi Pemakai	87
4.6. Format File Gambar	88
4.7. Penjelasan Program	88
4.7.1. Program Utama	88
4.7.2. Fungsi SetKeGrafik	94
4.7.3. Fungsi Kursor	96
4.7.4. Fungsi Cetak_Koord	96
4.7.5. Fungsi Load_Menu	98
4.7.6. Fungsi DefPoint	99
4.7.7. Fungsi PilihMenu	105
4.7.8. Fungsi Tampil_Menu	110
4.7.9. Fungsi Menu_2_Titik	112
4.7.10. Fungsi Restore	113
4.7.11. Fungsi PilihMenuLain	113
4.7.12. Fungsi Set_Garis	115
4.7.13. Fungsi Set_Fill	115
4.7.14. Fungsi Tentukan_Titik_Ke_2	116
4.7.15. Fungsi Kotak_Putus	118
4.7.16. Fungsi Cetak_Tanda	118
4.7.17. Fungsi Simpan_Status	118
4.7.18. Fungsi Restore_Status	118
V. PENUTUP	119
5.1. Kesimpulan	122
5.2. Saran	122

DAFTAR PUSTAKA	125
APPENDIKS	127

DAFTAR GAMBAR

Gambar	Halaman
2.1. Diagram Blok Pada Sistem Board IBM PC	8
2.2. Register-Register Pada 8088	10
2.3. Konfigurasi Slot IBM PC-XT	17
2.4. Pengalamatan Port I/O Pada CPU 8088	21
2.5. Peta Alamat Port I/O	22
2.6. Alamat Port I/O Yang Digunakan Oleh Main Board	23
2.7. Alamat Port I/O Yang Digunakan Oleh Slot .	24
2.8. Metoda Pengalamatan Memori 8088	25
2.9. Peta Memori IBM PC	26
2.10. Beberapa Versi Koordinat	37
2.11. Sistem Koordinat Polar	38
2.12. Pemilihan Pixel Pada Algoritma Garis	43
2.13. Pemilihan Pixel Pada Algoritma Lingkaran .	47
2.14. Pixel Setelah Direfleksi 8 Kali	48
2.15. Pemilihan Pixel Pada Algoritma Fill	53

2.16. Metoda Analog to Digital Converter	59
2.17. Flash ADC	62
2.18. Diagram Blok Successive Approximation ADC	64
3.1. Diagram Blok Perangkat Keras	68
3.2. Rangkaian Enable	71
3.3. Rangkaian Lengkap Board Write Strobe	73
3.4. Rangkaian Lengkap Port Output Pada Alamat 697	74
3.5. Rangkaian Lengkap Board Read Strobe	76
3.6. Rangkaian Lengkap Port Input Pada Alamat 696 & 697	77
3.7. Rangkaian Interface Lengkap	78
3.8. Rangkaian ADC	81
3.9. Rangkaian Lengkap	82
4.1. Diagram Blok Program	85
4.2. Konsep Trigonometrinya	93
4.3. Flowchart Fungsi CETAK_KOORD()	98
4.4. Flowchart Fungsi DEFPOINT()	101
4.5. Lanjutan Ke-1 Flowchart Fungsi DEFPOINT()	102
4.6. Lanjutan Ke-2 Flowchart Fungsi DEFPOINT()	103
4.7. Flowchart Fungsi TAMPIL_MENU()	111
4.8. Flowchart Fungsi PILIHMENU LAIN()	114
4.9. Flowchart Fungsi TENTUKAN_TITIK_KE_2	117

DAFTAR TABEL

Tabel	Halaman
2.1. Mode Video	29
2.2. Resolusi CGA	30
2.3. Warna Pada IBM PC	31
2.4. Kode Warna Atribut	32
2.5. Palette Standard Untuk Mode 6	33
2.6. Palette 0 Untuk Mode 4 dan 5	34
2.7. Palette 1 Untuk Mode 4 dan 5	34
5.1. Hasil Pengukuran	121

BAB I

PENDAHULUAN

1.1. Latar Belakang

Memang tepat apa yang diungkapkan oleh suatu ungkapan yaitu suatu gambar akan lebih berharga daripada seribu kata sekalipun. Mungkin tampaknya ungkapan tersebut agak berlebihan, namun bila dikaji lebih jauh ternyata memang benar, karena sebuah gambar lebih mampu menjelaskan banyak informasi secara padat dan tepat.

Karena itulah banyak orang lebih mengandalkan gambar daripada kata-kata. Mulai dari zaman batu, di mana orang belum mengenal tulisan, mereka sudah pandai mengungkapkan perasaan dengan gambar, hal ini bisa dibuktikan dengan adanya relief-relief.

Sampai saat ini, di jaman yang sangat modern ini, gambar masih memegang peranan penting.

Sejalan dengan perkembangan teknologi, cara menggambar pun mengalami kemajuan. Bila dulu orang lebih suka menggambar dengan pensil, kuas dan alat-alat

gambar yang lain, maka sekarang komputer pun sudah banyak dipakai orang untuk menggambar.

Sehingga muncullah disiplin ilmu yang baru, yaitu komputer grafik.

Manfaat demi manfaat pun mulai dirasakan dengan adanya komputer sebagai alat bantu gambar/disain (Computer Aided Design / CAD). Industri-industri pun tidak ketinggalan dalam memanfaatkan komputer untuk mendisain produk-produknya.

Untuk itulah perangkat lunak grafik sangat dibutuhkan. Selain itu, untuk pemanfaatan perangkat lunak grafik tersebut diperlukan juga suatu peralatan input yang interaktif sehingga akan memberi banyak kemudahan kepada pemakai dalam membuat suatu gambar.

1.2. Tujuan

Membuat suatu perangkat lunak grafik serta perangkat keras yang akan digunakan sebagai interface untuk menghubungkan papan grafik yang sudah tersedia dengan komputer IBM PC-XT.

Perangkat lunak dan perangkat keras ini dijalankan dan dihubungkan pada komputer IBM PC.

Dengan menggunakan perangkat keras bersama-sama dengan papan grafik yang tersedia maka pemakaian perangkat lunak grafik akan menjadi lebih mudah.

1.3. Permasalahan dan Pembatasannya

Untuk membuat program grafik yang mempunyai

kemampuan memadai diperlukan banyak fasilitas agar dapat dimanfaatkan oleh pemakai. Pada perangkat lunak grafik sebenarnya banyak fasilitas yang dibutuhkan. Namun untuk perangkat lunak grafik yang dibuat ini tidak disediakan semuanya.

Fasilitas grafik yang disediakan hanyalah terbatas pada pengeditan gambar dan penggambaran bentuk-bentuk tertentu. Sedangkan fasilitas yang tidak disediakan adalah memasukkan teks, fasilitas transformasi serta UNDO (pembatalan).

Yang termasuk dalam teknik transformasi adalah memutar gambar, translasi gambar, proyeksi gambar, penskalaan gambar (memperbesar dan memperkecil).

Untuk lebih jelasnya berikut ini akan disebutkan fasilitas-fasilitas yang disediakan dalam perangkat lunak grafik yang akan dibuat :

- Membuat garis
- Membuat lingkaran
- Membuat segiempat
- Meng-copy gambar yang telah diblok sebelumnya tanpa merubah gambar aslinya
- Meng-copy gambar yang telah diblok sebelumnya dan menghapus gambar aslinya
- Merubah warna gambar
- Merubah warna latar belakang
- Membersihkan layar
- Mencetak gambar ke printer

Permasalahan akan timbul tidak hanya dalam usaha mengadakan fasilitas-fasilitas tersebut namun juga dalam pengaturan memori agar seefisien mungkin. Hal ini diperlukan karena data grafik membutuhkan memori yang cukup besar.

Format dari file gambar adalah berdasarkan pada bit map, sehingga file gambar yang dihasilkan oleh perangkat lunak grafik ini mudah dikonversikan ke dalam format lain yang dimiliki oleh perangkat lunak grafik yang lain.

Selain itu untuk menyelesaikan tugas akhir ini, juga dibuat suatu perangkat keras yang berfungsi sebagai interface dengan komputer mikro.

Komputer mikro yang dipilih adalah komputer IBM PC-XT dengan pertimbangan bahwa komputer tersebut sudah semakin populer dan literatur-literatur yang membahas masalah komputer tersebut relatif mudah didapatkan.

Perangkat keras yang dibuat tersebut akan digunakan bersama-sama dengan papan grafik yang telah tersedia. Cara penggunaannya adalah dengan menggerak-gerakkan lengan bantu yang ada pada papan grafik. Posisi lengan grafik mencerminkan posisi titik yang dipilih yang ada pada papan grafik.

Prinsip yang digunakan oleh perangkat keras yang akan dibuat adalah mengkonversikan data posisi lengan bantu yang berupa sinyal analog menjadi data digital yang dapat diolah oleh komputer. Untuk itu digunakan rangkaian Analog to Digital Converter.

1.4. Sistematika Pembahasan

Susunan dan sistematika pembahasan di dalam buku tugas akhir ini dapat dijelaskan sebagai berikut :

- Bab dua membahas mengenai arsitektur IBM PC-XT, adapter tampilan dari IBM PC/XT dan segala permasalahannya, elemen-elemen grafik yang banyak digunakan dalam pembuatan suatu program grafik serta teknik-teknik rangkaian 'analog to digital converter'.
- Bab tiga membahas mengenai perencanaan dan pembuatan perangkat keras yang akan berfungsi untuk mengkonversikan data posisi lengan bantu pada papan grafik menjadi data digital yang kemudian dihubungkan pada komputer IBM PC-XT.
- Bab empat membahas mengenai perencanaan dan pembuatan program grafik. Selain itu juga dibahas kegunaan dari fungsi-fungsi yang penting.
- Bab lima membahas uji coba perangkat lunak serta pengukuran perangkat keras.
- Bab enam merupakan bab terakhir yang membahas kesimpulan dari isi buku tugas akhir ini serta usulan-usulan yang perlu dipertimbangkan untuk pengembangan lebih lanjut.

BAB II

TEORI PENUNJANG

Dalam bab ini akan dijelaskan mengenai teori penunjang dari alat yang direncanakan dan akan dibuat. Alat yang berupa interface ini dibuat agar bisa berhubungan dengan komputer mikro IBM. Oleh sebab itu terlebih dahulu tentulah harus dijelaskan bagaimana arsitektur dari mikrokomputer IBM.

Setelah mengenal arsitekturnya maka kemudian akan dijelaskan sedikit tentang bagaimana caranya mikrokomputer IBM mampu menghasilkan suatu tampilan grafik.

Dan yang tak kalah penting untuk dijelaskan adalah dasar-dasar komputer grafik, karena di samping akan dibuatkan suatu interface (berupa perangkat keras) juga akan dibuatkan suatu program komputer grafik yang akan memanfaatkan kemampuan dari interface tersebut.

Berikutnya akan dijelaskan metoda-metoda yang biasa digunakan pada Analog to Digital Converter (A/D Converter). Ini perlu dijelaskan karena interface yang

dibuat nanti akan menggunakan rangkaian A/D Converter.

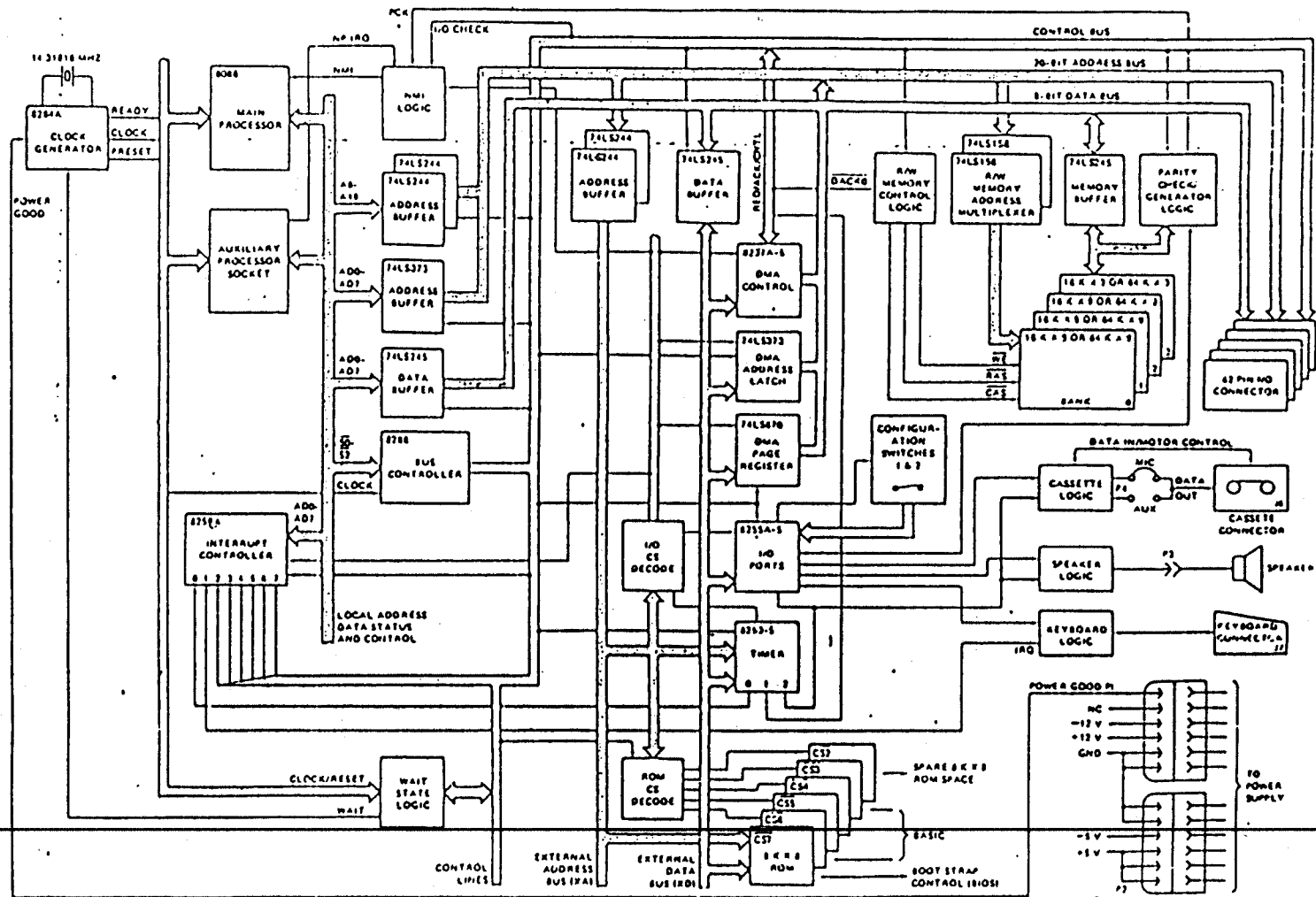
2.1. Arsitektur Mikrokomputer IBM

Mikrokomputer IBM menggunakan mikroprosesor 8088 produksi Intel sebagai Central Processing Unitnya (CPU). Mikroprosesor 8088 ini termasuk generasi 16 bit yang laris di pasaran setara dengan larisnya mikrokomputer IBM. Hal ini disebabkan mikroprosesor ini memiliki kemampuan yang lebih baik jika dibandingkan dengan mikroprosesor 8 bit yang terdahulu, terutama kemampuan pengalamatan data serta sinyal-sinyal kontrol yang lebih banyak serta didukung oleh instruksi-instruksi dalam bahasa mesin yang lebih banyak, sehingga akan sangat memudahkan pembuatan perangkat lunak.

Sistem Unit IBM terdiri dari mikroprosesor 16 bit 8088, Read Only Memory, Random Access Memory, Power Supply serta sebuah speaker untuk aplikasi audio ditambah 5 buah ekspansi slot (8 buah untuk IBM PC-XT) untuk pengembangan sistem board lebih lanjut, termasuk untuk keperluan Expansion Card Controller seperti Disk Drive Controller, Printer Card Controller, Monochrome / Colour Graphic Card, Serial Port Controller dan lain-lain.

System Unit Processor Board IBM terdiri atas :

- Mikroprosesor 8088 sebagai CPU
- Rangkaian Clock
- Sistem Bus
- Sistem ROM



GAMBAR 2.1. DIAGRAM BLOK PADA SYSTEM BOARD IBM PC¹⁾

1) Hall, Douglas V., Microprocessors and Interfacing: Programming and Hardware, Mc. Graw Hill Book Co., Singapore, 1986, hal 369.

- Sistem RAM
- Sistem Timer / Counter
- Sistem DMA
- Sistem Interrupts
- Sistem Adapter

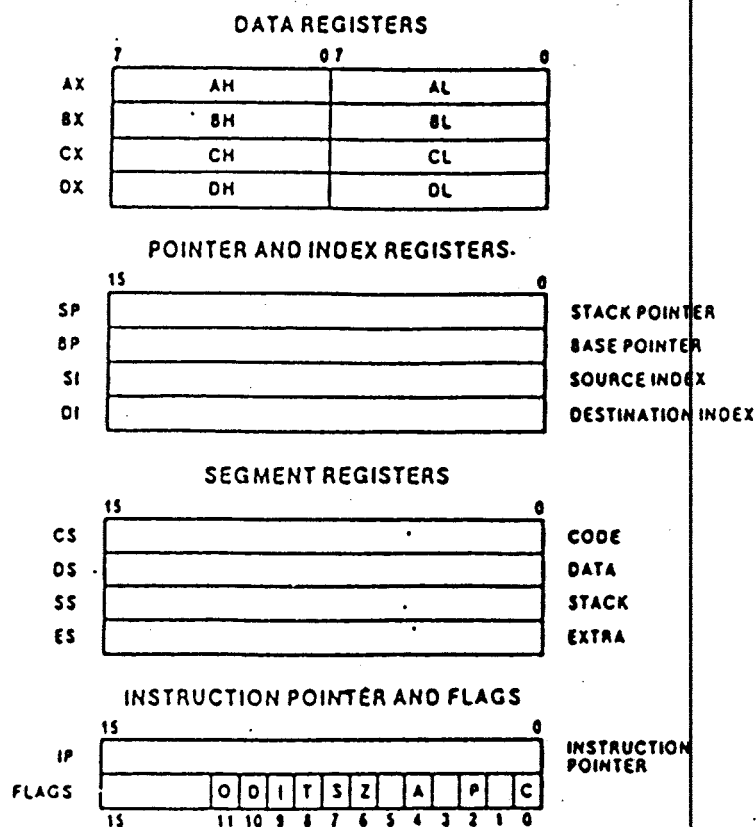
Hubungan antar bagian-bagian yang ada pada System Unit Processor Board IBM bisa dilihat pada gambar 2.1.

2.1.1. Mikroprosesor 8088

CPU 8088 merupakan otak dari Sistem Board IBM. CPU 8088 merupakan mikroprosesor 16 bit yang memiliki 8 bit memory bus. Walaupun demikian instruksi-instruksinya dapat memanipulasi data 16 bit. Akan tetapi data dan instruksi-instruksinya diambil dan ditulis ke memori 8 bit satu per satu pada siklus tertentu. Mikroprosesor ini mampu mengakses memori sampai 1 megabyte, baik berupa data maupun program. Sebenarnya mikroprosesor 8088 ini merupakan turunan dari mikroprosesor 8086, hanya berbeda pada lebar data bus. Mikroprosesor 8086 mempunyai 16 bit internal data path maupun eksternal data bus, sedangkan mikroprosesor 8088 mempunyai internal data path 16 bit tetapi memory data bus hanya 8 bit. Tetapi walaupun demikian perangkat lunak untuk keduanya sama persis, hanya berbeda pada kecepatan eksekusi saja pada pengaksesan data 16 bit.

Mikroprosesor 8088 ini memiliki 14 buah register 16 bit yang terbagi atas 4 buah Data Register, 4 buah

Pointer & Indeks Register, 4 buah Segment Register serta Instruction Pointer dan Flags Register seperti yang tampak pada gambar 2.2.



GAMBAR 2.2.

REGISTER - REGISTER PADA 8088²⁾

Mikroprosesor 8088 memiliki bus alamat dengan lebar 20 bit yang memungkinkan untuk mengakses 1.048.576

2) Eggebrecht, Lewis C., Interfacing to the IBM Personal Computer, Howard W. Sams & Co., Indianapolis, 1987.

lokasi memori. Tetapi karena instruksi-instruksinya hanya mengizinkan operasi dan manipulasi alamat 16 bit, maka seolah-olah hanya 65.536 lokasi memori saja yang dapat diakses. Untuk mengatasi hal tersebut maka mikroprosesor 8088 dilengkapi dengan register khusus yang dinamakan segment register (16 bit), yang memungkinkan diaksesnya 16 segmen memori yang tiap segmennya terdiri dari 65.536 lokasi.

Mikroprosesor 8088 mempunyai empat macam segment register, yaitu :

- Code Segment : Segment Register yang menunjukkan segment dari program / instruksi-instruksi dari program / opcode.
- Data Segment : Segment Register yang menunjukkan segment dari variabel-variabel / array / data dari program.
- Stack Segment : Segment Register yang menunjukkan segment dari semua data yang diakses oleh SP (Stack Pointer) atau BP (Base Pointer).
- Extra Segment : Segment Register tambahan yang dapat berhubungan baik dengan stack, data ataupun code segment.

2.1.2. Rangkaian Clock

Mikrokomputer IBM menggunakan clock 4,77 MHz untuk menjalankan mikroprosesor 8088. Sebenarnya

rangkaian oscillator yang digunakan di-drive oleh kristal 14,31818 MHz. Melalui 8284A Clock Chip frekwensi tersebut dibagi tiga menjadi 4,77 MHz untuk kemudian dihubungkan ke 8088. Selanjutnya Clock 4,77 MHz ini masih dibagi lagi menjadi 4 sehingga menghasilkan clock 1,19 MHz yang digunakan untuk men-drive clock input dari sistem timer / counter.

2.1.3. Sistem Bus

Bagian-bagian fungsional dari Sistem Board IBM dihubungkan ke mikroprosesor 8088 melalui Sistem Bus. Bus ini terdiri atas bermacam-macam sinyal antara lain : Data Bus, Address Bus, Control, Timing, Interrupt Request & DMA Control. Di samping itu Sistem Board dilengkapi pula dengan 8288 Bus Controller Chip yang menghasilkan sinyal-sinyal kontrol seperti MEMR, MEMW, IOR, IOW dan lain-lain.

2.1.4. Sistem ROM

Sistem Board IBM dilengkapi dengan 64 KByte Read Only Memory (ROM). ROM ini di-decode sedemikian rupa sehingga menempati 64 KB lokasi memori teratas dalam peta memori IBM. ROM ini di-decode mulai lokasi F0000 - FFFFF. Adapun ROM ini berisi program bahasa mesin / Assembly 8088 yang antara lain berfungsi untuk :

- Inisialisasi sistem
- Power On Diagnostic & System Checkout
- System Configuration Determination

- I/O Device Driver yang disebut BIOS yang mengatur sistem keyboard, video, disk drive, printer
- Diskette Bootstrap Loader
- Font Bit Pattern dari character ASCII

Sedangkan sisa lokasi ROM berisi perangkat lunak Microsoft BASIC Interpreter yang menempati lokasi sekitar 32 KByte.

2.1.5. Sistem RAM

Sistem Board IBM dilengkapi dengan Random Access Memory (RAM) yaitu memori yang dapat dibaca maupun ditulisi, yang menempati lokasi memori 00000 - 3FFFF (256 KB) dan dapat diperluas sampai lokasi 9FFFF (640 KB). Sistem Board RAM ini sebenarnya terdiri atas 9 bit data, tetapi bit yang kesembilan hanyalah merupakan parity bit yang dihasilkan pada setiap siklus penulisan memori (memory write cycle) dan diisikan pada bit kesembilan dari sistem Dynamic RAM. Pada pembacaan ulang data dari RAM, parity bit diregenerasi lagi dari kedelapan bit data tersebut kemudian dibandingkan dengan parity bit yang semula disimpan pada parity RAM. Jika kedua parity bit tersebut sama, berarti data yang terbaca dan yang tertulis sebelumnya adalah benar (valid). Lokasi memori yang terendah dari sistem board RAM ini digunakan oleh DOS (Disk Operating System) yang antara lain digunakan untuk Interrupt Vector, DOS Data

Area, DOS Communication Area sedangkan sisanya digunakan untuk keperluan pemakai.

2.1.6. Sistem Timer / Counter

Untuk mendukung sistem timing dan fungsi counting, mikrokomputer IBM dilengkapi dengan sebuah 8253-5 Timer / Counter Chip dengan 16 bit Timer Counter. Input clock dari ketiga channel chip ini di-drive oleh sinyal clock 1,19 MHz. Output Timer channel 0 dihubungkan ke Sistem Interrupt level 0 & diprogram untuk menghasilkan interrupt setiap 54,925 milidetik yang digunakan oleh Sistem I/O Routine & Sistem Time-of-Day Clock / Calender. Output Timer channel 1 menghasilkan DMA Request pada DMA channel 0 yang digunakan untuk me-refresh System Dynamic Memory untuk menghasilkan Dummy Read Cycle setiap 72 clock prosesor atau setiap 15,12 mikrodetik. Output Timer channel 2 digunakan untuk men-drive System Audio Speaker.

2.1.7. Sistem DMA

Beberapa peralatan I/O seperti Disk Drive, mentransmisikan data lebih cepat daripada yang dapat dilakukan oleh prosesor lewat kontrol program. Untuk menangani hal ini maka Sistem Board IBM dilengkapi dengan chip 8237-5 DMA Controller. Chip ini dapat mentransmisikan data antara memori dan interface / adapter tanpa memerlukan kontrol dari prosesor, dengan demikian proses transmisi data berlangsung sangat cepat

sehingga dapat menghemat waktu proses. Chip Controller ini memiliki 4 channel yang memiliki fungsi dan urutan prioritas sebagai berikut :

- Channel 0 : memory refresh RAM (prioritas tertinggi)
- Channel 1 : belum digunakan
- Channel 2 : digunakan oleh Disk Drive Adapter
- Channel 3 : belum digunakan (prioritas terendah)

2.1.8. Sistem Interrupt

Mikroprosesor 8088 mempunyai 2 buah fasilitas interrupt, yaitu Maskable Interrupt & Non-Maskable Interrupt. Tetapi karena lebih banyak level interrupt yang diperlukan maka Sistem Board IBM dilengkapi dengan chip 8259 Interrupt Controller yang memperluas kemampuan Maskable Interrupt sampai 8 level Interrupt. Dari kedelapan level tersebut, 2 level telah digunakan oleh Sistem Board yaitu level 0 (tertinggi) digunakan untuk menerima interrupt dari Timer / Counter channel 0. Level 1 digunakan untuk menerima interrupt dari keyboard. Sedangkan sisa 6 level lainnya tersedia pada Sistem Bus dan dapat digunakan untuk keperluan lain. NMI Interrupt dari mikroprosesor digunakan untuk keperluan Base-Board RAM Parity, I/O Channel Check & 8087 Numeric Processor.

2.1.9. Sistem Adapter

Sistem Board IBM dilengkapi dengan beberapa adapter seperti keyboard, audio speaker, audio cassette.

Peralatan ini terintegrasi pada sistem bus melalui paralel register chip I/O 8255A-5 yang mempunyai 3 buah 8 bit I/O Port. Port-port 8255A-5 ini digunakan antara lain untuk membaca data dari konfigurasi dip-switch (Memory Select Function), membaca scan-code dari keyboard, men-drive audio speaker, membaca serial data dari audio cassette, mengontrol motor, audio cassette machine dll.

Mengenai detail dari arsitektur IBM PC-XT ini dapat dilihat dengan lebih jelas pada buku Technical Reference Manual - XT System.

2.2. Slot Pada IBM PC-XT

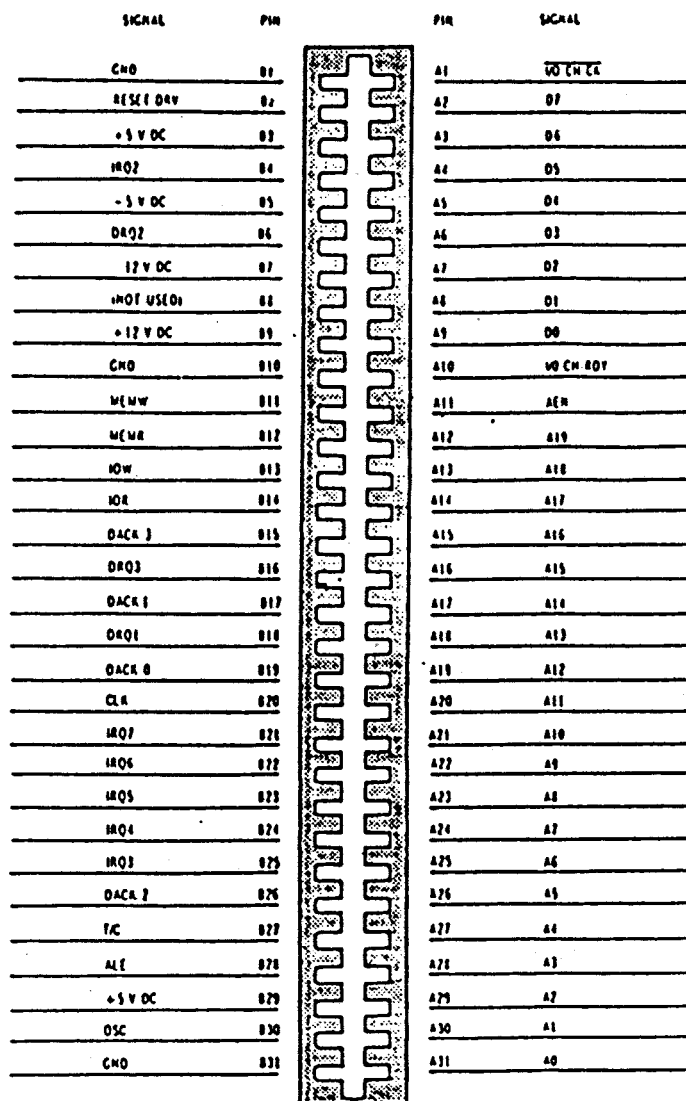
Sistem board IBM PC-XT mempunyai 8 buah slot (I/O Channel). Semua slot tersebut mempunyai konfigurasi pin yang sama dan dapat mengakses memori dan I/O Map yang sama. Melalui slot inilah alat yang direncanakan akan berhubungan dengan IBM PC-XT.

Slot IBM PC-XT merupakan konektor dengan 62 pin yang konfigurasinya dapat dilihat pada gambar 2.3. Dalam perencanaan alat, tidak semua pin dari slot tersebut digunakan, melainkan hanya beberapa pin saja.

Berikut ini penjelasan dari masing-masing pin serta penggunaannya :

- OSC & CLK : Line-line ini hanya mampu menghasilkan sinyal saja, tanpa mampu menerima sinyal. Sinyal yang dihasilkan adalah sinyal clock dengan frekwensi masing-masing

14.31818 MHz dan 4.77 MHz seperti telah dijelaskan sebelumnya.



GAMBAR 2.3.

KONFIGURASI SLOT IBM PC-XT³⁾

- RESET DRV : Line ini digunakan untuk mereset atau menginisialisasikan sistem logika selama

3) *ibid*, hal 77.

power-up.

- A0 - A19 : Merupakan line-line yang hanya mampu menerima sinyal.

Line-line ini digunakan untuk mengakses memori dan peralatan I/O dari sistem. Kedua puluh address line ini dapat mengakses sampai 1MB lokasi memori. Line-line ini bisa dihasilkan oleh prosesor 8088 maupun DMA Controller.

- D0 - D7 : Line-line ini digunakan sebagai jalur data, baik oleh prosesor, memori atau peralatan I/O. D0 merupakan LSB (Least Significant Bit) sedangkan D7 merupakan MSB (Most Significant Bit).

- ALE : Address Latch Enable.

Line ini dihasilkan oleh 8288 Bus Controller dan menandakan bahwa bus alamat sudah valid serta digunakan oleh System Board untuk me-latch (mengunci) alamat yang benar dari prosesor. Line ini tidak akan diaktifkan selama siklus DMA.

- IRQ2-IRQ7 : Interrupt Request 2 sampai 7

Line-line ini berhubungan langsung dengan 8259 Interrupt Controller dan dipakai oleh unit I/O untuk meng-interrupt CPU 8088.

- IOR : Input / Output Read.

Line ini digunakan untuk menandai bahwa

bus cycle yang terjadi adalah cycle untuk membaca data dari port I/O dan alamat yang berada pada bus alamat adalah alamat dari port I/O. Kemudian port I/O pada alamat tersebut akan menghasilkan respon dengan menempatkan datanya pada data bus.

- IOW : Input / Output Write.

Line ini digunakan untuk menandai bahwa bus cycle yang terjadi adalah cycle untuk membaca data yang ada pada data bus untuk kemudian menuliskannya pada port I/O.

- MEMR : Memory Read.

Line ini digunakan untuk menginstruksikan memori agar meletakkan datanya pada data bus. Line ini aktif low dan dapat dibangkitkan baik oleh prosesor maupun oleh DMA Controller.

- MEMW : Memory Write.

Line ini digunakan untuk menginstruksikan memori agar menyimpan data yang berada pada data bus.

- AEN : Address Enable.

Line ini digunakan untuk meng-non-aktifkan kontrol prosesor serta membiarkan DMA Controller untuk mengambil alih operasi. Jika line ini aktif (High)

maka DMA Controller telah mengambil alih kontrol atas address bus, data bus, perintah pembacaan (dari memori atau I/O port) maupun perintah penulisan (ke memori atau I/O port).

- Bus power dan ground :

Pada slot tersedia 4 jenis tegangan yaitu +5 volt, -5 volt, +12 volt, -12 volt.

Besarnya tegangan di atas diukur terhadap ground. Besar daya power supply yang dipakai pada umumnya sekitar 63 watt.

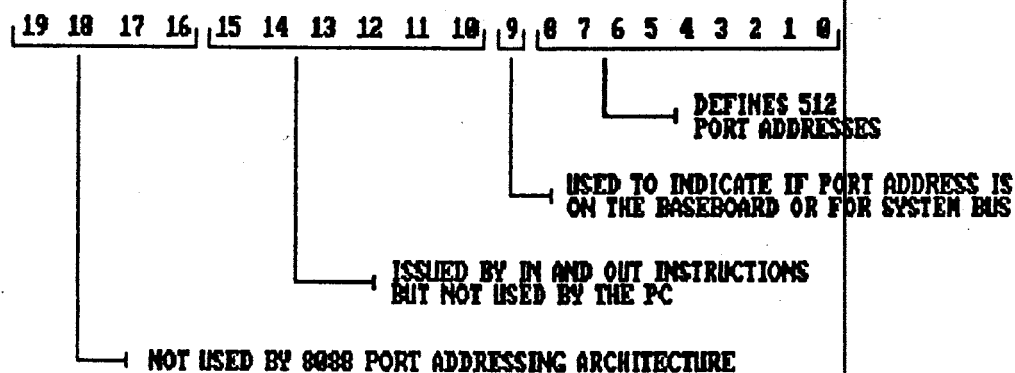
2.3. Pengalamatan Port I/O

Peralatan pelengkap dan I/O adapter pada PC dikontrol oleh input dan output port digital. Pengalamatan port-port ini menggunakan ruang alamat I/O port yang ada pada CPU 8088. Arsitektur CPU 8088 mampu menangani 65.536 alamat port. Disain PC biasanya tidak menggunakan seluruh alamat port tersebut. Pengalamatan port-port ini hanya menggunakan 10 bit dari 16 bit yang tersedia. Jadi, hanya bit 0 sampai bit 9 yang di-decode untuk alamat peralatan atau port, sehingga tersedia 1024 alamat port.

Bit 9 pada alamat port I/O memiliki arti khusus pada disain PC. Pada saat bit ini tidak aktif, maka sistem bus tidak dapat menerima data dari slot yang ada di sistem board. Hanya data dari peralatan atau I/O port yang ada di sistem board yang dapat diterima oleh sistem

bus. Namun pada saat bit 9 aktif, maka data dari slot akan dapat diterima oleh sistem bus. Berarti, dari 1024 alamat port input yang ditangani oleh PC bisa dibagi menjadi 2 bagian yaitu 512 alamat port berada di system board dan 512 alamat port yang lainnya berada di slot.

Tapi, yang penting diketahui adalah pembagian port di atas hanya berlaku untuk port input. Sedangkan untuk port output seluruh alamat port (1024 buah) yang tersedia bisa digunakan.



GAMBAR 2.4.

PENGALAMATAN PORT I/O PADA CPU 8088⁴⁾

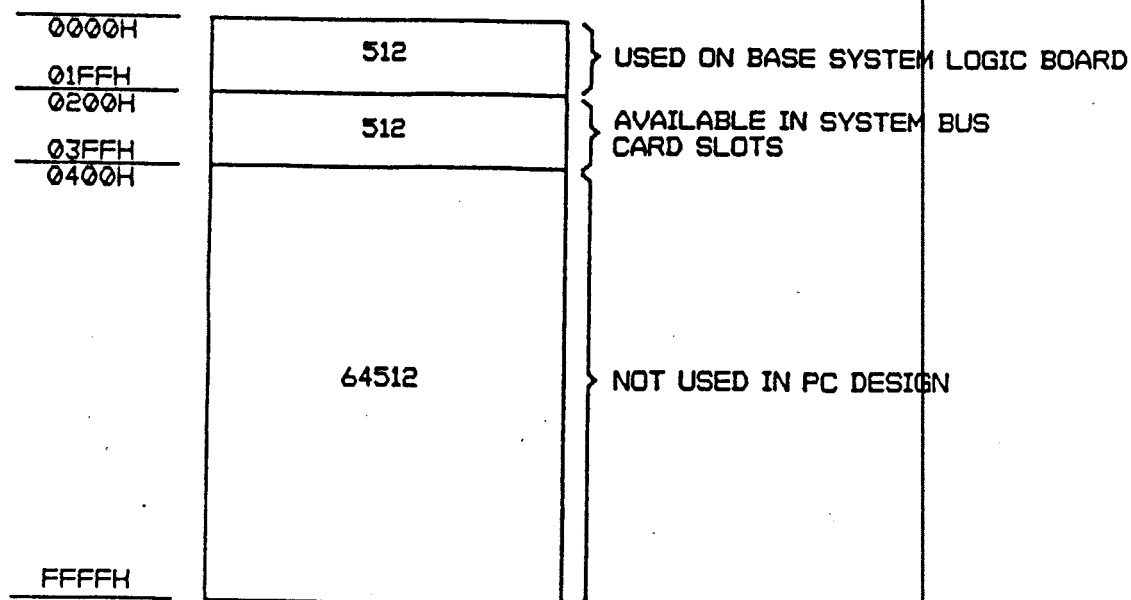
2.3.1. Penetaan Alamat Port I/O

Peta alamat port I/O dapat dibagi menjadi 2 bagian. Bagian pertama adalah dengan alamat heksa 0000 sampai 01FF. Bagian ini digunakan untuk alamat-alamat dari peralatan dan I/O terpadu yang ada pada sistem

4) Ibid, hal 126.

board induk.

Bagian kedua dari port I/O memiliki alamat mulai dari heksa 0200 sampai 02FF. Ruang alamat ini digunakan untuk alamat port yang berada pada slot sistem board. Untuk lebih jelasnya pemetaan alamat port I/O ini bisa dilihat pada gambar 2.5.



GAMBAR 2.5.

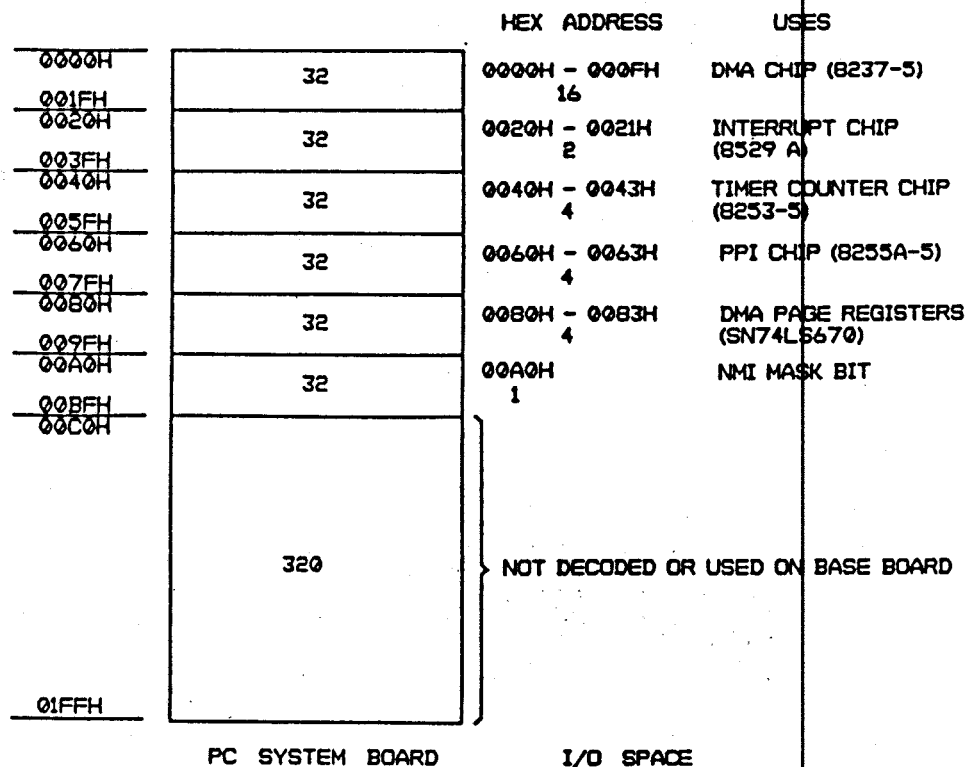
PETA ALAMAT PORT I/O⁵⁾

Dalam merencanakan alat, bisa digunakan alamat-alamat port yang tidak dipakai oleh peralatan-peralatan yang lain sehingga tidak akan terjadi crash data antara alat yang dibuat dengan peralatan-peralatan tersebut.

2.4. Penetaan Memori

IBM PC-XT mempunyai 20 bit bus alamat, maka

5) Ibid, hal 127.



GAMBAR 2.6.

ALAMAT PORT I/O YANG DIGUNAKAN OLEH MAIN BOARD⁶⁾

berarti dapat mengakses lokasi sebanyak 1 MByte atau tepatnya 1.048.576 byte. Hal ini dapat dicapai dengan memanipulasi 16 bit bus alamat 8088 dengan metoda segmen register seperti diperlihatkan pada gambar 2.8.

Untuk dapat memberi alamat lokasi memori sebesar 1 Megabyte diperlukan 2 buah register. Register pertama digunakan untuk mengamati offset sedangkan register lainnya digunakan untuk mengamati segmen-nya. Cara

⁶⁾ Ibid, hal 128.

		HEX ADDRESS	USES
0200H	1	0200H	(NOT USED)
0201H	1	0201H	GAME CONTROL ADAPTER
0202H	118	0202H - 0277H	(NOT USED)
0277H			
0278H	8	0278H - 027FH	SECOND PRINTER PORT ADAPTER
027FH			
0280H	120	0280H - 02F7H	(NOT USED)
02F7H			
02F8H	8	02F8H - 02FFH	SECOND SERIAL PORT ADAPTER
02FFH			
0300H	120	0300H - 0377H	(NOT USED)
0377H			
0378H	8	0378H - 037FH	PRINTER PORT ADAPTER
037FH			
0380H	48	0380H - 03AFH	(NOT USED)
03AFH			
03B0H	16	03B0H - 03BFH	MONOCHROME AND PRINTER ADAPTER
03BFH			
03C0H	16	03C0H - 03CFH	(NOT USED)
03CFH			
03D0H	16	03D0H - 03DFH	COLOR/GRAPHICS ADAPTER
03DFH			
03E0H	16	03E0H - 03EFH	(NOT USED)
03EFH			
03F0H	8	03F0H - 03F7H	5.25 DISK DRIVE ADAPTER
03F7H			
03F8H	8	03F8H - 03FFH	SERIAL PORT ADAPTER
03FF			

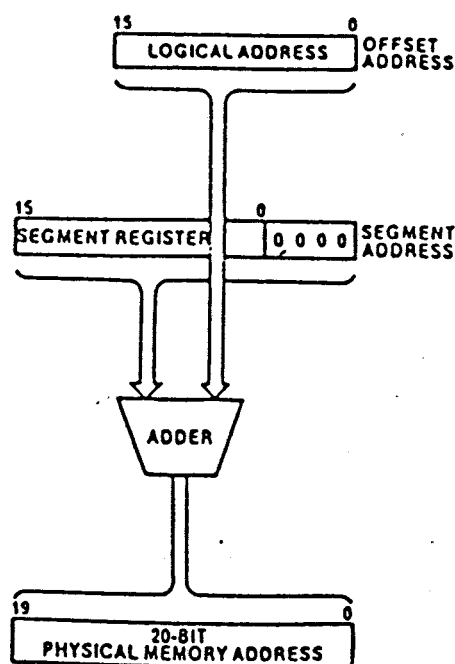
GAMBAR 2.7.

ALAMAT PORT I/O YANG DIGUNAKAN OLEH SLOT⁷⁾

pengalamatannya adalah dengan menggeser register segment ke kiri 4 kali kemudian ditambahkan dengan register offset.

Dalam disain IBM PC-XT membagi dua bagian lokasi memori yaitu bagian atas dan bagian bawah. Sebagai

7) Ibid, hal 129.



GAMBAR 2.8.

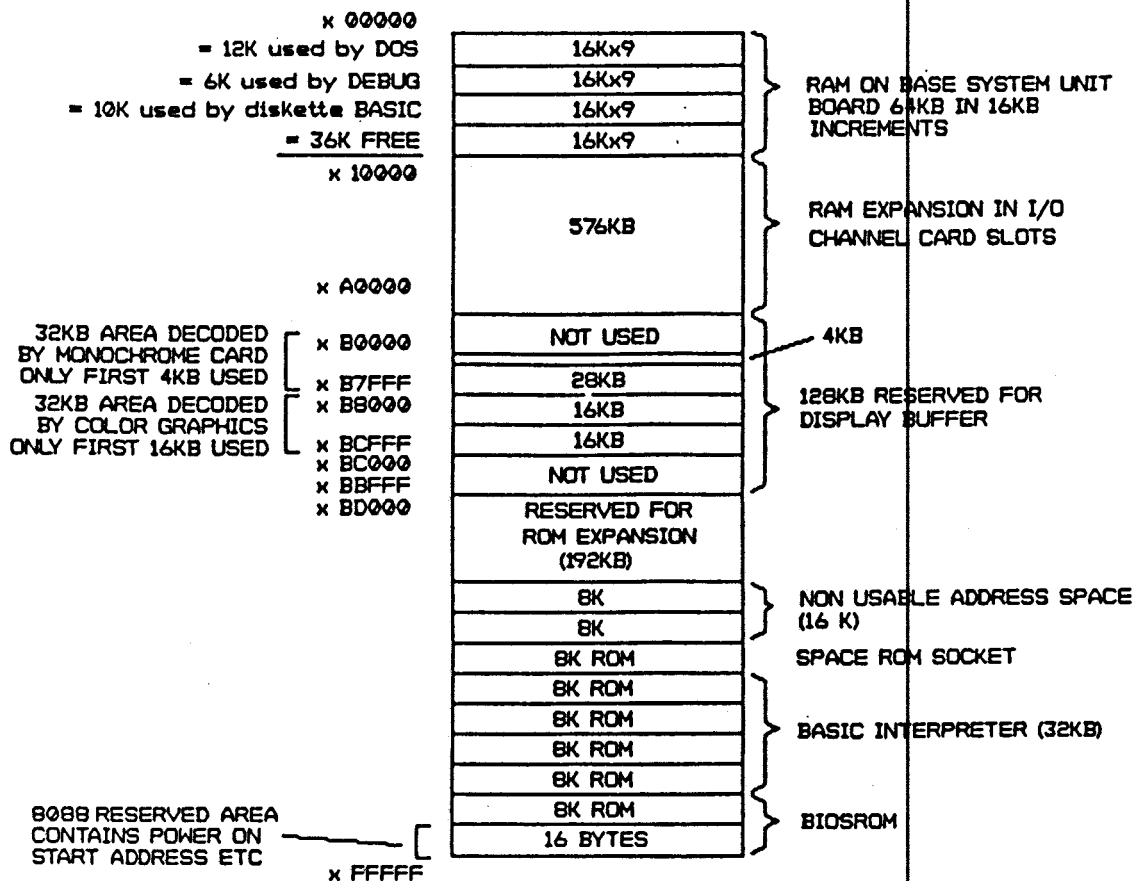
METODA PENGALAMATAN MEMORI 8088⁸⁾

tambahan display adapter juga menggunakan alamat memori ini sebagai penyangga display. Pada bagian atas alamat memori ditempati oleh ROM. Dari 64 KiloByte yang disediakan pada ruang bagian atas diisi ROM sebanyak 40 KiloByte. ROM ini terletak pada board induk dan tidak dapat diletakkan pada sistem bus slot.

RAM ditempatkan pada awal ruang bagian bawah memori untuk PC yaitu sebanyak 64 KiloByte. Sedangkan untuk XT berisikan 256 KiloByte atau 640 KiloByte pada board induk. Untuk tambahan dapat digunakan RAM ekspansi yang ditancapkan pada slot.

8) Ibid, hal 34.

Display adapter mengambil tempat 32 KiloByte yang digunakan sebagai penyangga display. Color Graphic Card menggunakan 16 KiloByte dari 32 KiloByte yang disediakan. Monochrome Display Adapter menggunakan 4 KiloByte dari 32 KiloByte yang disediakan.



GAMBAR 2.9.

PETA MEMORI IBM PC⁹⁾

9) Ibid, hal 139.

Disain IBM PC-XT menyediakan tempat kosong dari sistem memori untuk pengembangan lebih lanjut. Pemetaan memori di atas bisa dilihat pada gambar 2.9.

2.5. Tampilan Pada Komputer IBM PC

2.5.1. Adapter Tampilan

Untuk menghasilkan tampilan, komputer IBM PC dan keluarganya membutuhkan suatu adapter tampilan. Adapter tampilan ini menghubungkan komputer dengan layar monitor melalui sebuah chip yang disebut CRT controller. Pada adapter ini juga terdapat I/O port yang bisa diprogram, ROM pembangkit karakter dan RAM untuk menangani informasi tampilan.

Terdapat banyak jenis adapter tampilan, tetapi semuanya diproduksi setelah munculnya 2 jenis adapter, yaitu : Color Graphics Adapter dan Monochrome Adapter.

Tampilan video dihasilkan oleh 2 mode dasar yang berbeda yang disebut mode teks dan mode grafik.

Mode teks hanya mampu menghasilkan karakter teks, sedangkan mode grafik selain mampu menghasilkan gambar yang kompleks juga mampu menghasilkan karakter teks dalam berbagai bentuk dan ukuran.

Color Graphics Adapter mampu dioperasikan pada kedua mode, baik teks maupun grafik. Sedangkan Monochrome Adapter hanya bisa dioperasikan pada mode teks saja.

2.5.1.1. Memori dan Adapter Tampilan

Memori untuk tampilan secara fisik ditempatkan pada kartu adapter bersama-sama rangkaian tampilan lainnya. Namun secara logika, memori tampilan ini merupakan bagian dari memori utama komputer.

Untuk memori tampilan disediakan 128 K byte yaitu pada blok A dan B dengan alamat mulai dari A0000 sampai BFFFF.

Monochrome Adapter membutuhkan 4 K byte memori dengan alamat awal B000, dan Color Graphics Adapter membutuhkan 16 K byte dengan alamat awal B800. Sedangkan yang lainnya (yang belum terpakai) yaitu mulai A000 sampai B000 dicadangkan untuk pemakaian tampilan dengan teknik yang lebih tinggi dari MA dan CGA, misalnya: EGA (Enhanced Graphics Adapter).

2.5.1.2. Menampilkan ke Layar

Baik Monochrome maupun Color Graphics Adapter menyimpan informasi tampilan pada "memory mapped display" (tampilan yang terpetakan pada memori).

Mengapa disebut demikian ? Sebab, setiap alamat pada memori tampilan berhubungan dengan lokasi tertentu di layar. Jadi dengan meletakkan data byte langsung pada memori tampilan sama artinya dengan menampilkan data tersebut pada layar.

2.5.2. Mode Tampilan Video

Pada mulanya hanya terdapat 8 mode video pada IBM

PC. Namun sekarang telah bertambah yaitu dengan 7 atau lebih mode yang merupakan pengembangan dari 8 sebelumnya. Tiap mode video tersebut memberikan batasan karakteristik tampilan, jumlah teks yang bisa ditampilkan, resolusi serta warna tampilan yang disediakan.

TABEL 2.1.
MODE VIDEO¹⁰⁾

Mode	Type	Size	Colors	Adapter	Display
0	Text	40 × 25	16 (grey)	CGA, EGA, PCjr	Enhanced Color
1	Text	40 × 25	16 foreground, 8 background	CGA, EGA, PCjr	Enhanced Color
2	Text	80 × 25	16 (grey)	CGA, EGA, PCjr	Enhanced Color
3	Text	80 × 25	16 foreground, 8 background	CGA, EGA, PCjr	Enhanced Color
4	Graphics	320 × 200	4	CGA, EGA, PCjr	Enhanced Color
5	Graphics	320 × 200	4 (grey)	CGA, EGA, PCjr	Enhanced Color
6	Graphics	640 × 200	2	CGA, EGA, PCjr	Enhanced Color
7	Text	80 × 25	b/w	EGA, MA	Monochrome
8	Graphics	160 × 200	16	PCjr	Enhanced Color
9	Graphics	320 × 200	16	PCjr	Enhanced Color
10	Graphics	640 × 200	4	PCjr	Enhanced Color
11	Apparently internal to the EGA				
12	Apparently internal to the EGA				
13	Graphics	320 × 200	16	EGA	Enhanced Color
14	Graphics	640 × 200	16	EGA	Enhanced Color
15	Graphics	640 × 350	b/w	EGA	Monochrome
16	Graphics	640 × 350	64	EGA	Enhanced Color

2.5.2.1. Resolusi Tampilan

Gambar grafik dibentuk dari beberapa dot, yang disebut elemen gambar (picture element) atau pixel.

10) Norton, Peter, Programmer's Guide to The IBM PC, Washington, Microsoft Press, 1985, hal 73.

Resolusi tampilan ditentukan oleh jumlah baris, dari atas sampai bawah layar, dan jumlah kolom, dari kiri sampai kanan layar.

Jumlah baris yang bisa ditampilkan oleh layar monitor ditentukan oleh perangkat keras dan sinyal video. IBM PC yang standard biasanya hanya mampu menampilkan 25 baris teks dan 200 baris grafik.

Pada mode grafik terdapat 2 resolusi untuk Color Graphics Adapter, yaitu medium dan high.

TABEL 2.2.
RESOLUSI CGA¹¹⁾

Resolusi	Pixel	Characters
Medium	320x200	40x25
High	640x200	80x25

2.5.3. Penakaaian Warna

Terdapat banyak warna yang bisa ditampilkan pada berbagai mode tampilan kecuali untuk Monochrome Adapter.

Warna-warna untuk layar tampilan PC merupakan kombinasi dari 4 elemen yaitu 3 komponen warna (merah, hijau dan biru) serta satu komponen intensitas atau kecermelangan.

Mode teks dan grafik sama-sama memakai warna dan intensitas namun satu dengan yang lain memiliki cara

11) Ibid, hal 74.

yang berbeda dalam mengkombinasikannya.

Mode teks yang lebih menitik-beratkan pada karakter yang dibentuk oleh beberapa pixel, membutuhkan satu byte data untuk menset warna, intensitas, karakter kedip serta latar belakang.

TABEL 2.3.
WARNA PADA IBM PC¹²⁾

Intensity Red Green Blue	Number	Description
0 0 0 0	0	Black
0 0 0 1	1	Blue
0 0 1 0	2	Green
0 0 1 1	3	Cyan (blue-green to civilians)
0 1 0 0	4	Red
0 1 0 1	5	Magenta
0 1 1 0	6	Brown (or dark yellow)
0 1 1 1	7	Light grey (or ordinary white)
1 0 0 0	8	Dark grey (black on many screens)
1 0 0 1	9	Light blue
1 0 1 0	10	Light green
1 0 1 1	11	Light cyan
1 1 0 0	12	Light red
1 1 0 1	13	Light magenta
1 1 1 0	14	Yellow (or light yellow)
1 1 1 1	15	Bright white

Bila dipilih mode dengan 16 warna maka akan diperoleh warna 0 - 15, untuk mode 8 warna akan diperoleh warna 0 - 7 yang meliputi seluruh warna namun tanpa komponen intensitas. Untuk mode dengan 4 warna maka akan didapatkan 4 buah warna yang dipilih dari 16 warna yang ada, 4 warna ini disebut palette.

12) Ibid, hal 76.

Sedangkan pada mode dengan 2 warna hanya akan didapatkan warna hitam dan putih saja.

2.5.3.1. Warna Pada Mode Teks

Dalam mode teks untuk Color Graphics Adapter seluruh warna yang ada bisa digunakan. Untuk latar depan bisa memanfaatkan 16 buah warna, namun untuk latar belakang hanya dapat menggunakan 8 warna.

Di dalam mode teks, setiap karakter yang ada pada layar tampilan diatur oleh 2 byte informasi di memori. Byte pertama berisi kode ASCII untuk karakter yang akan ditampilkan, dan byte kedua mengatur bagaimana karakter tersebut akan ditampilkan. Hal ini berhubungan dengan warnanya. Byte kedua ini disebut atribut karakter. Atribut karakter teks ini terdiri dari 3 komponen, yaitu latar depan (warna karakter itu sendiri), latar belakang, dan komponen kedip (blink).

TABEL 2.4.
KODE WARNA ATRIBUT¹³⁾

Bit 7 6 5 4 3 2 1 0	Use
1	Blinking of foreground character
. 1	Red component of background color
. . 1	Green component of background color
. . . 1 . . .	Blue component of background color
. . . . 1 . .	Intensity component of foreground color
. 1 .	Red component of foreground color
. 1	Green component of foreground color
. 1	Blue component of foreground color

13) Ibid, hal 80.

2.5.3.2. Warna Pada Mode Grafik

Di atas telah dijelaskan bagaimana caranya memberi warna karakter pada mode teks. Untuk mode grafik, teknik yang dipakai sangatlah berbeda dari mode teks. Pada mode grafik setiap pixel yang ada pada layar memiliki warna yang menjadikan pixel itu nampak di layar.

Pada saat menulis teks dalam mode grafik, maka satu warna diberikan pada pixel-pixel yang menjadi latar belakang dan satu warna lagi diberikan pada pixel-pixel yang menjadi latar depan (membentuk karakter). Warna yang diberikan pada suatu pixel dalam mode grafik ini adalah warna yang terdapat pada palette mode tersebut.

TABEL 2.5.

PALETTE STANDARD UNTUK MODE 6¹⁴⁾

Bit	Value	Color
0 0	0	Black
0 1	1	White

2.6. Dasar-Dasar Komputer Grafik

2.6.1. Sistem Koordinat

Di dalam komputer grafik, sebelum titik, garis dan elemen-elemen lain yang lebih kompleks dapat ditempatkan pada layar tampilan, maka perlu dibahas

14) Ibid, hal 82.

TABEL 2.6.
PALETTE 0 UNTUK MODE 4 DAN 5¹⁵⁾

Bit	Value	Color
0 0	0	Black (default; may be changed to any color)
0 1	1	Green
1 0	2	Red
1 1	3	Brown

TABEL 2.7.
PALETTE 1 UNTUK MODE 4 DAN 5¹⁶⁾

Bit	Value	Color
0 0	0	Black (default; may be changed to any color)
0 1	1	Cyan
1 0	2	Magenta
1 1	3	Normal white

bagaimana caranya menentukan lokasi suatu titik pada layar.

Untuk menentukan lokasi suatu titik dibutuhkan suatu sistem koordinat. Koordinat suatu titik adalah angka (skalar) yang mendefinisikan titik tersebut dengan acuan titik asal.

Sistem koordinat yang biasa dipakai adalah sebagai berikut:

- a. Cartesian (rectangular)
- b. Polar

15) Ibid, hal 83.

16) Norton, loc cit.

2.6.1.1. Koordinat Cartesian (Rectangular)

Sistem koordinat yang paling sederhana dan paling populer adalah sistem koordinat cartesian. Sistem koordinat ini memiliki sumbu horizontal dan sumbu vertikal. Lokasi layar ditentukan oleh koordinat (x, y) . Nilai koordinat itu sendiri merupakan banyaknya spasi baik pada sumbu horizontal (x) maupun vertikal (y) . Jumlah spasi ini dihitung dari titik asal sistem tersebut yaitu $(0, 0)$. Sistem koordinat ini bisa juga mengandung nilai negatif.

Koordinat cartesian dipakai dalam sistem grafik digital sebab dapat disesuaikan dengan baik ke permukaan tampilan yang berbentuk raster. Raster adalah kisi-kisi posisi yang teratur yang bisa ditempati oleh informasi grafik. Permukaan tampilan pada komputer merupakan salah satu contoh raster.

Sistem digital biasanya berdasarkan raster, yang berarti bahwa memori dipetakan pada tampilan sebagai suatu garis horizontal di atas garis horizontal yang lain. Nilai yang ada dalam memori adalah biner, di mana nilai inilah yang mengatur status pixel pada tampilan.

2.6.1.1.1. Kelinieran (Linieritas)

Kelinieran merupakan salah satu karakteristik dari sistem koordinat cartesian. Pergerakan sejauh 3 satuan dalam arah x , sebagai contoh, akan memiliki efek yang sama, tidak peduli di mana gerakan itu terjadi.

Ciri lain yang perlu diketahui adalah tidak tergantung pada sumbu yang lain. Gerakan pada sumbu x tidak akan menyebabkan terjadinya gerakan pada sumbu y , begitu sebaliknya. Secara matematis sumbu-sumbu tersebut ortogonal satu sama lain.

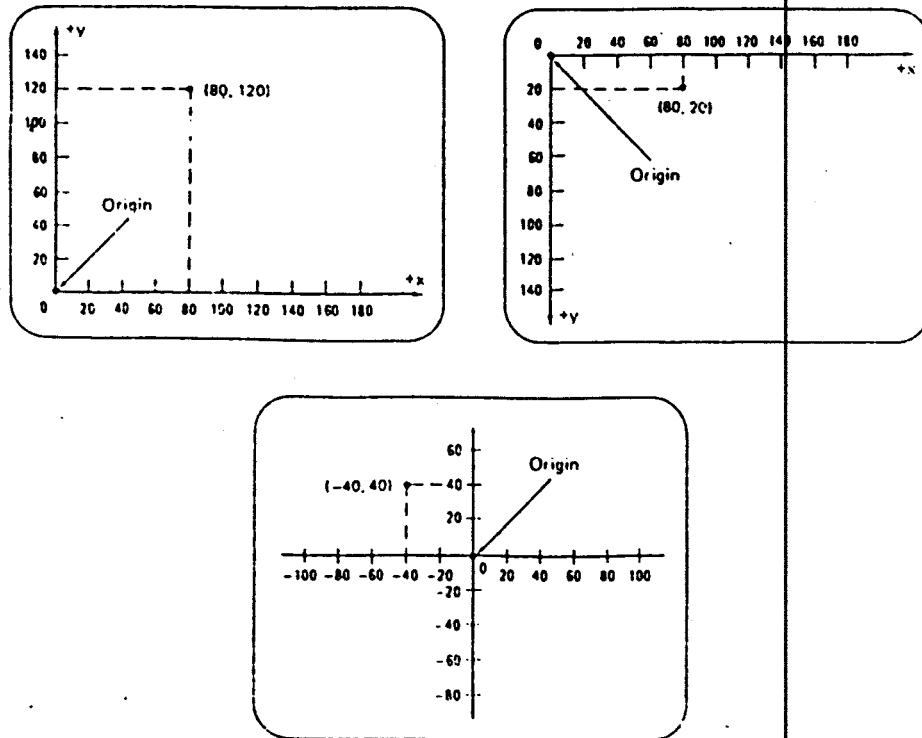
2.6.1.1.2. Arah dan Titik Asal

Sementara sudah adanya anggapan bahwa sistem koordinat cartesian adalah cara paling baik untuk menentukan posisi titik di layar dalam banyak aplikasi, namun masih sering terjadi pertentangan yang berkisar pada arah dan titik asalnya. Ahli-ahli matematika, statistik dan orang-orang bisnis beranggapan bahwa titik awalnya berada pada ujung kiri bawah layar dengan arah x positif ke kanan dan arah y positif ke atas layar.

Namun orang-orang teknik berpendapat bahwa titik asal berada di sudut kiri atas dari layar sedangkan arah x positif dan y positif berturut-turut adalah ke kanan dan ke bawah layar. Hal ini sehubungan dengan operasi secara fisik dari peralatan tampilan pada umumnya. Orang-orang yang bekerja dengan grafik 3-D lebih suka menganggap bahwa titik asal berada pada pusat layar sehingga membentuk sistem koordinat 4 kuadran. Ini timbul sehubungan dengan dibutuhkannya nilai koordinat negatif pada aplikasi grafik 3-D. Penyelesaian dua masalah di atas diserahkan sepenuhnya kepada pemakai untuk menentukan yang mana yang akan dipilih.

2.6.1.2. Koordinat Polar

Sistem koordinat polar kadang-kadang juga dipakai



GAMBAR 2.10.

BEBERAPA VERSI KOORDINAT CARTESIAN¹⁷⁾

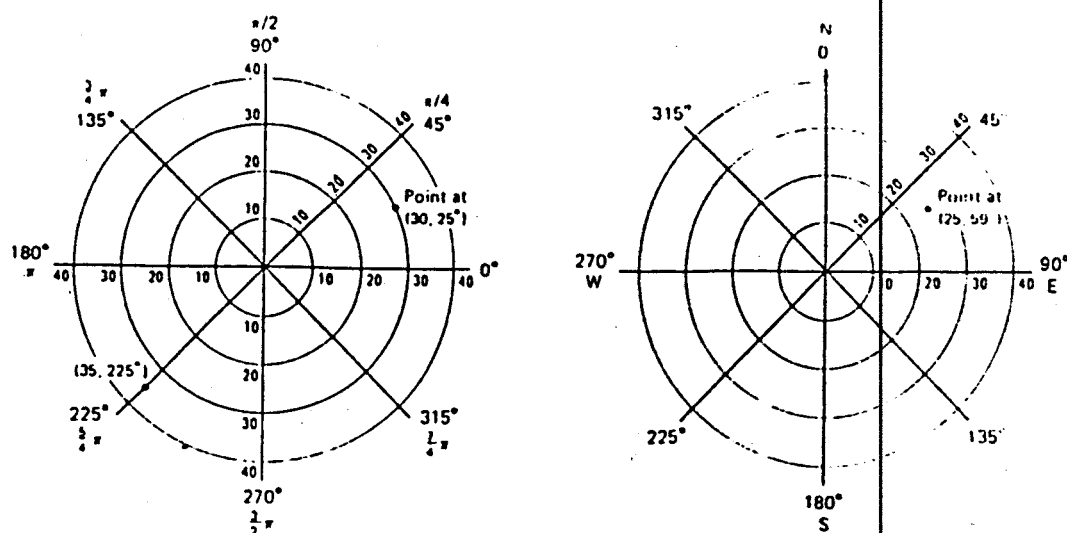
dalam komputer grafik, tetapi bila dibandingkan dengan sistem koordinat cartesian kurang dapat diandalkan dalam hal penentuan lokasi pixel pada tampilan.

Koordinat polar diberikan oleh sudut dan jarak. Dalam suatu sistem koordinat polar, lokasi dari suatu titik ditentukan oleh defleksi angular suatu garis. Titik-titik akhir suatu garis adalah titik asal dan lokasi dari titik itu sendiri. Koordinat polar diberi

17) Artwick, Bruce A., Microcomputer Displays, Graphics, and Animation, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1985, hal 11.

lambang theta dan rho. Theta menunjukkan sudut, sedang rho adalah radius atau panjang.

Sama seperti koordinat cartesian, ada juga beberapa parameter yang dipertentangkan dalam koordinat



GAMBAR 2.11.

SISTEM KOORDINAT POLAR¹⁸⁾

polar ini. Mengenai titik asal biasanya ditempatkan pada pusat layar, namun arah sudut positif bisa dipilih searah atau berlawanan arah dengan jarum jam. Selain itu titik dengan sudut 0 derajat bisa horizontal (di kanan) atau vertikal (di atas).

Untuk mewujudkan gambar grafik pada suatu sistem koordinat, maka banyak elemen yang harus diperhatikan, seperti titik, garis, kurva, lingkaran dan sebagainya. Karena adanya perbedaan antara dunia analog tempat kita

18) *Ibid*, hal 13.

berada dan dunia digital dari komputer, maka ada beberapa proses yang khusus dibutuhkan dalam penggambaran elemen-elemen tersebut pada layar komputer.

2.6.2. Titik

Dalam geometri analitis, suatu titik dianggap tidak memiliki dimensi dan ukuran. Titik hanya menunjukkan suatu lokasi dalam suatu sistem koordinat. Dalam sistem grafik, titik-titik ditunjukkan oleh nilai numerik dari koordinat-koordinatnya. Dalam sistem 2 dimensi, suatu titik ditunjukkan oleh 2 koordinat. Dalam sistem 3 dimensi, suatu titik ditunjukkan oleh 3 koordinat. Dan untuk sistem koordinat yang lebih dari 3 dimensi, misalnya n dimensi, maka suatu titik ditunjukkan oleh n koordinat.

Secara teknis, suatu titik tidak memiliki algoritma. Dia hanya memiliki lokasi. Dalam sistem koordinat 2 dimensi, lokasi dari suatu titik terdiri dari 2 angka skalar biasanya disebut x & y . Penggambaran suatu titik pada layar tampilan sangatlah tergantung pada metoda operasi tampilan tersebut. Namun pada umumnya, fungsi untuk menggambar titik merupakan proses perubahan warna pixel yang ada pada permukaan tampilan. Salah satu contoh instruksi yang ada pada Turbo C V. 2.0 untuk menggambar suatu titik :

```
putpixel(x, y, warnapixel)
```

Pada sistem hitam putih warna pixel digantikan oleh parameter lain yaitu tingkat kecoklatan (gray

level).

Beberapa peralatan display biasanya memperbolehkan pemakai untuk membaca status dari titik yang ada di layar. Fasilitas ini sangat bermanfaat bagi pembuatan program grafik.

Contoh instruksi untuk membaca warna suatu titik pada Turbo C V. 2.0 :

```
getpixel(x, y)
```

2.6.3. Lokasi Titik

Dengan memikirkan satu titik sebagai suatu lokasi tanpa dimensi maka akan banyak memberikan bantuan dalam menyelesaikan masalah-masalah yang timbul dalam pemrograman grafik.

2.6.4. Garis

Mungkin kita masih ingat bahwa suatu garis dalam geometri didefinisikan sebagai jarak terpendek antara 2 titik. Ini adalah definisi dari Euclidean.

Garis memiliki satu dimensi, jadi dia hanya memiliki panjang saja.

Garis adalah orde terendah dari obyek grafik yang memiliki dimensi dan bisa dikatakan ada dalam grafik. Pada saat suatu garis digambar, maka sebenarnya yang digambar adalah pixel-pixel yang menutupinya. Suatu fungsi penggambaran titik dipakai untuk menjadikan pixel-pixel tersebut on atau off.

2.6.5. Kurva

Kurva adalah kelas yang umum dalam geometri, termasuk di dalamnya adalah lingkaran, ellips, busur dan plot-plot dari persamaan lainnya. Untuk tujuan pemrograman grafik, kurva lebih tepat didefinisikan sebagai suatu kumpulan titik.

2.6.6. Fill

Fill adalah suatu daerah warna, entah sana entah tidak, yang dibatasi oleh garis atau kurva. Fill bisa diselesaikan dalam banyak cara yang berbeda, tergantung pada tipe database yang telah dirancang sebelumnya.

Fill bisa dilakukan secara flood, di mana seluruh pixel yang berada pada daerah yang diinginkan diset sesuai dengan warna yang diinginkan. Atau bisa juga diselesaikan dengan cara memberi garis silang (cross hatching) pada daerah yang ditentukan. Penyilangan ini dilakukan sampai pada batas daerah yang telah ditentukan.

2.6.7. Algoritma Garis

Karena keberadaan suatu titik hanyalah menunjukkan lokasinya, titik tersebut sebenarnya tidak pernah di-plot pada layar. Suatu pixel, yang merupakan unit terkecil dari informasi pada tampilan harus memiliki dimensi fisik dalam suatu sistem digital. Dimensi fisik

ini berupa nilai on atau off.

Titik yang ditunjukkan pada permukaan tampilan sebenarnya adalah suatu pixel yang diisi oleh definisinya sendiri. Bisa juga dianggap sebagai suatu garis dengan panjang satu pixel.

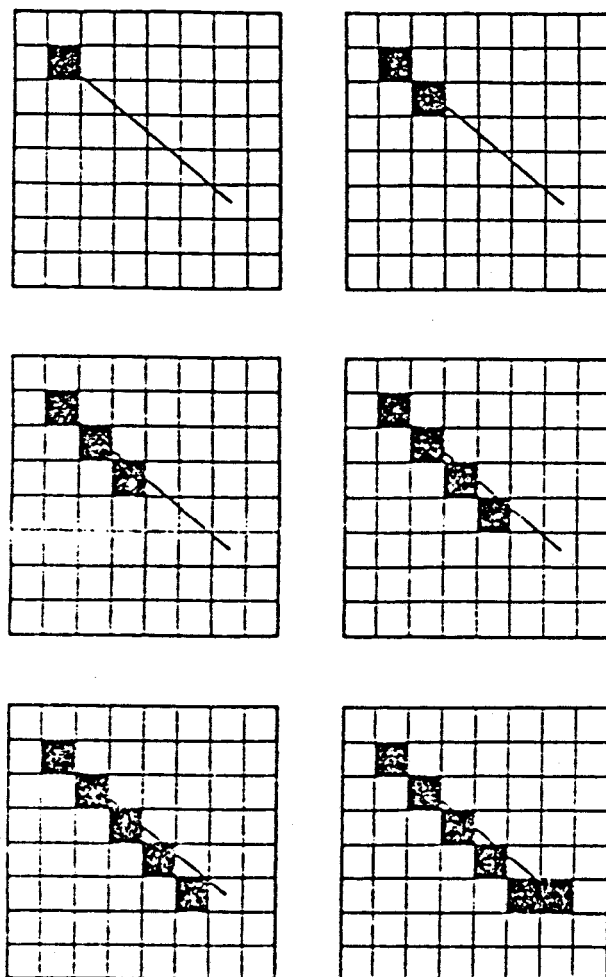
Garis dibuat dari beberapa titik. Lebih tepatnya, garis dibentuk oleh pixel-pixel yang menutupinya.

Suatu program untuk menggambar garis adalah sangat berguna dalam suatu sistem grafik. Kompleksitas dari program tersebut tergantung pada kecepatan penggambaran garis yang diinginkan dan kemampuan sistem tampilannya. Oleh sebab itu untuk membuat program penggambar garis yang efisien maka diperlukan algoritma yang tepat.

Kunci untuk mengerti algoritma grafik adalah dengan membandingkan antara dunia analog dan digital. Pensil di tangan dapat dipakai untuk menggambar garis lurus dengan bantuan penggaris. Garis yang dihasilkan akan berupa garis mulus, karena garis ini adalah garis analog. Namun untuk menggambar garis yang sama dengan menggunakan sistem digital maka harus di-plot pada pixel-pixel yang paling dekat dengan garis analog tadi. Algoritma untuk memilih pixel-pixel mana saja yang terdekat dengan garis analog tadi akan menjadikan garis tersebut tidak bisa di-plot secara mulus. Akan terjadi perpatahan pada sistem raster.

2.6.7.1. Raster

Bentuk raster tidaklah mulus, karena dibuat dari beberapa pixel yang diskrit dan masing-masing memiliki dimensi. Tergantung pada seberapa besar setiap pixel, maka suatu garis yang digambar dalam raster akan dibagi dalam beberapa patahan.



GAMBAR 2.12.

PEMILIHAN PIXEL PADA ALGORITMA GARIS¹⁹⁾

19) Johnson, Nelson, Advanced Graphics In C, Programming and Techniques, Mc. Graw Hill Inc., Berkeley, California, 1987, hal 73.

Konsep dari rasterisasi serupa dengan pembulatan pada matematika.

2.6.7.2. Algoritma Bresenham Untuk Garis

Algoritma yang paling populer untuk menggambar garis adalah algoritma Bresenham.

Dasar Dari algoritma ini adalah penyimpanan besarnya kesalahan antara lokasi ideal dan lokasi pixel yang sebenarnya. Besar kesalahan akan ditambah dan diperbandingkan secara berulang-ulang (iterasi). Setiap kali nilai x dan y ditambah untuk mendapatkan nilai x dan y yang sebenarnya. Kemudian, pixel di-plot pada lokasi yang sebenarnya yaitu pada koordinat (x, y) .

2.6.7.3. Pola Garis

Garis yang dibuat dengan algoritma di atas tidak hanya dalam bentuk padat, tapi berbagai pola garis bisa dibuat, misalnya garis putus-putus, garis dash-dot dan sebagainya.

Cara yang dilakukan biasanya dengan menggunakan bilangan integer tidak bertanda yang berisi bit pola garis. Pola ini di-plot berulang-ulang sepanjang garis.

2.6.7.4. Lebar Garis

Lebar garis bisa divariasi dengan jalan mem-plot kembali garis tersebut pada beberapa offset. Untuk itu diperlukan fungsi tambahan yang memanggil fungsi garis. Dengan satu pengertian, lebar suatu garis bisa disamakan

dengan segiempat yang diisi dengan satu warna.

Algoritma di atas bisa juga dipakai, namun proses penggambaran garis menjadi lebih lamban. Cara yang lebih baik adalah dengan menggambar garis berikutnya di samping yang sudah ada, sejauh satu pixel.

Tipe variasi dari penggambaran garis ini lebih kompleks untuk diwujudkan daripada variasi pola garis.

Beberapa masalah yang terutama sekali dalam aspek lebar garis ini adalah bagaimana menangani titik-titik akhir dari garis yang lebar yang saling bertemu. Bila pada ujung akhirnya berupa segiempat, maka akan terlihat seperti pita yang bertemu pada ujungnya.

Teknik yang sangat kompleks memang telah dikembangkan untuk persoalan garis-garis yang saling bertemu ini.

2.6.7.5. Menghilangkan Patahan Garis (Antialiasing)

Proses antialias pada suatu garis digunakan untuk mengurangi patahan-patahan yang ada pada garis tersebut. Teknik antialias tergantung seberapa banyak warna yang disediakan pada suatu palette atau seberapa banyak tingkat kecoklatan yang disediakan.

Untuk tampilan dengan resolusi yang relatif kecil, alias akan sangat mengganggu.

Tampilan yang cukup bisa dipakai untuk mendapatkan hasil antialiasing yang baik dimulai pada resolusi 640 x 480. Dan bila ingin mendapatkan hasil yang betul-

betul mulus maka dibutuhkan tampilan dengan resolusi 1024 x 1024.

Untuk resolusi rendah, maka efek alias bisa dikurangi dengan menghitung jumlah pixel yang terpotong oleh garis. Jika sedikit pixelnya maka intensitas warna dikurangi secara proporsional. Cara lain untuk anti alias garis adalah dengan menambah pixel ke dalam tepi-tepi garis yang berpatahan. Metoda ini memberikan hasil garis yang mulus namun garis tersebut akan menjadi lebih lebar.

Dengan mengkombinasikan 2 teknik di atas, maka garis bisa dibuat lebih mulus lagi.

2.6.8. Algoritma Lingkaran

Lingkaran bisa dibuat dengan mem-plot garis sebagai suatu chord. Chord adalah segmen garis di mana titik-titik akhirnya berada pada keliling lingkaran. Tetapi lingkaran yang dihasilkan dengan chord cenderung tidak mulus.

Masalah lain pada metoda chord adalah banyaknya perhitungan yang diperlukan untuk menghasilkan lingkaran dari chord menyebabkan kecepatan penggambaran menjadi turun. Untuk mengatasinya maka perlu dibuat suatu algoritma yang mirip dengan algoritma garis. Salah satunya adalah dengan menggunakan algoritma Bresenham untuk lingkaran.

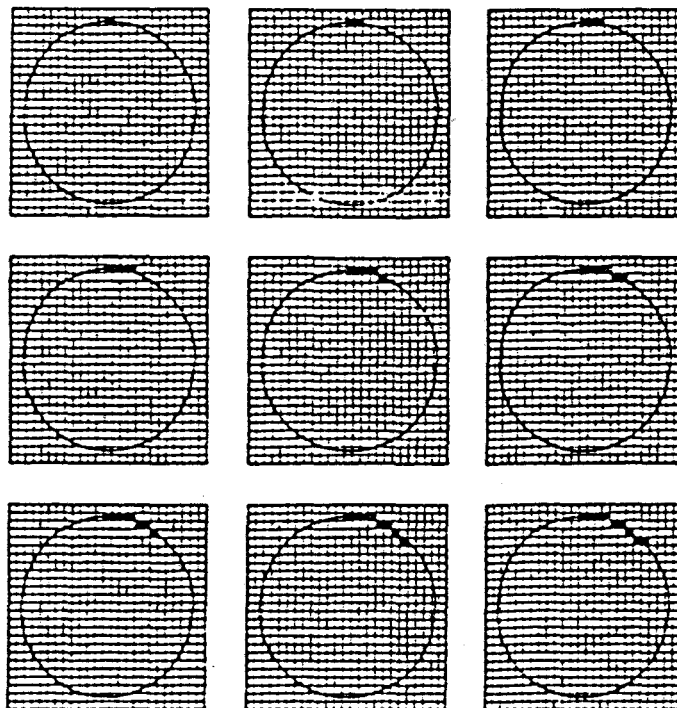
2.6.8.1. Algoritma Bresenham Untuk Lingkaran

Seperti pada garis, suatu lingkaran bisa dihasilkan dengan penambahan dan pengurangan nilai kumulatif.

Lokasi dari tiap pixel diperoleh dari nilai d (diameter), yang bervariasi tergantung pada x, y dan beberapa konstanta

2.6.8.2. Pola Lingkaran

Sama seperti pola garis, lingkaran bisa juga digambar dengan berbagai pola. Tekniknya juga sama dengan yang dipakai untuk garis.



GAMBAR 2.13.

PEMILIHAN PIXEL PADA ALGORITMA LINGKARAN²⁰⁾

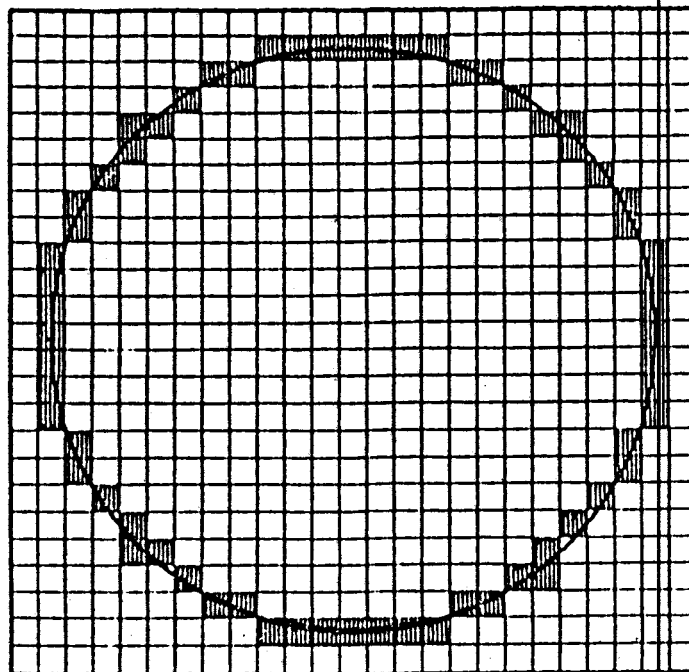
20) Ibid, hal 81.

2.6.8.3. Lebar Lingkaran

Garis dari lingkaran bisa dilebarkan dengan menggambar satu lingkaran lagi di dalam lingkaran yang sudah ada dan kemudian di sebelah luarnya. Untuk itu diperlukan suatu fungsi untuk memanggil fungsi penggambar lingkaran.

2.6.9. Busur

Proses penggambaran busur bisa juga dilakukan seperti pada lingkaran, namun akan ditemui beberapa kesulitan. Busur, yang merupakan bagian dari lingkaran



GAMBAR 2.14.

PIXEL SETELAH DIREFLEKSI 8 KALI²¹⁾

21) Ibid, hal 82.

hanya perlu digambar sudut sebagian. Sudut yang dilingkupi oleh busur adalah sudut yang dibutuhkan oleh suatu radius untuk menghasilkan busur.

Masalah bermula pada saat busur akan digambar mulai dari titik awal dan harus berakhir pada titik akhir. Titik awal dan titik akhir tidak bisa ditemukan pada busur. Oleh sebab itu sebelumnya harus ditentukan dulu arah penggambaran, yaitu bisa searah atau berlawanan arah jarum jam.

Dalam menggambar busur ini maka lebih baik menggunakan teknik penggambaran chord daripada dengan metoda pixel. Algoritma Bresenham yang dipakai untuk lingkaranpun tidak akan secepat teknik chord. Karena dipakainya metoda oktan dalam algoritma Bresenham, maka sangatlah kompleks untuk memilih oktan yang dibutuhkan untuk membuat suatu busur.

2.6.9.1. Lingkaran Dengan 3 Titik

Suatu lingkaran atau busur bisa didefinisikan dengan menempatkan titik pusatnya dan menentukan radiusnya. Bisa juga, lingkaran didefinisikan dengan menentukan lokasi 3 titik pada keliling lingkaran.

Metoda seperti ini sangatlah berguna dalam aplikasi teknik, sebab titik pusat dari lingkaran atau busur biasanya tidak diketahui.

Dari 3 buah koordinat yang telah diperoleh maka bisa dicari 3 buah persamaan simultan. Dengan menyelesaikan 3 persamaan simultan itu maka diperoleh

titik pusat dari lingkaran tersebut. Dan dengan mengurangi salah satu koordinat dengan titik pusat maka bisa dicari radius dari lingkaran, sehingga untuk selanjutnya lingkaran pun bisa digambar.

misalnya : koordinat titik ke 1 : (x_0, y_0)

koordinat titik ke 2 : (x_1, y_1)

koordinat titik ke 3 : (x_2, y_2)

$$x_{20} = x_0^2 - x_1^2$$

$$x_{n0} = 2x_0 - 2x_1$$

$$y_{20} = y_0^2 - y_1^2$$

$$y_{k0} = 2y_0 - 2y_1$$

$$x_{21} = x_1^2 - x_2^2$$

$$x_{n1} = 2x_1 - 2x_2$$

$$y_{21} = y_1^2 - y_2^2$$

$$y_{k1} = 2y_1 - 2y_2$$

$$x_{2y20} = x_{20} + y_{20}$$

$$x_{2y21} = x_{21} + y_{21}$$

$$x_{nyn} = x_{2y20}$$

$$y_{kn} = y_{k0}$$

$$x_{2y20} = x_{2y20} * x_{h1}$$

$$y_{k0} = y_{k0} * x_{h1}$$

$$x_{2y21} = x_{2y21} * x_{h0}$$

$$y_{k1} = y_{k1} * x_{h0}$$

$$y_c = (x_{2y20} - x_{2y21}) / (y_{k0} - y_{k1})$$

$$x_c = (x_{nyn} - y_{cykn}) / x_{h0}$$

$$x_0 = x_0 - x_c$$

$$y_0 = y_0 - y_c$$

$$r = \sqrt{x_0^2 + y_0^2}$$

Persamaan ini berpusat pada titik asal (0, 0)

2.6.10. Algoritma Fill

Walaupun kemampuan untuk menggambar garis dan kurva telah terpenuhi oleh fungsi penggambar garis dan kurva, namun sering pula diinginkan untuk mem-plot suatu daerah dengan warna. Untuk itu, bisa dilakukan dalam beberapa cara tergantung pada ketepatan, kecepatan dan fleksibilitas yang diinginkan. Salah satu metoda yang bisa dipakai adalah seperti di bawah ini.

Algoritma fill dipakai untuk mengubah warna pixel yang ada di dalam suatu daerah tertentu.

Kelihatannya mudah, namun bagi komputer hal ini tidak sesederhana yang diduga.

Algoritma fill yang sederhana dimulai dari pixel yang terletak dalam suatu daerah. Kemudian dicek warna pixel pada baris-baris di atasnya baru kemudian baris-baris di bawahnya. Biasanya, perbatasan yang menutupi suatu daerah berisi pixel dengan warna serupa. Perbatasan pada umumnya, berupa garis atau kurva yang berada pada bagian luar dari daerah tersebut.

2.6.10.1. Flood

Flood adalah daerah fill yang penuh. Flood dibedakan dari suatu fill berdasarkan kecepatannya. Masalah pada flood adalah apabila perbatasan yang ditentukan ternyata menjorok ke dalam daerah yang akan di-fill

sehingga terlihat seperti terpotong.

Bila algoritma yang dipakai sangat sederhana, maka akan didapat daerah yang tidak di-fill oleh warna. Salah satu cara untuk menyelesaikan masalah dari flood adalah dengan mempercayakan pada definisi obyek dan sistem database. Sekali obyek telah ditentukan maka batas-batasnya akan bisa dikenali. Bisa juga dilakukan pengecekan irisan, penyortiran dan berbagai teknik lain dengan menggunakan metoda recursive, inclusive dan lain-lain.

Algoritma yang bisa juga dipakai adalah dengan menentukan satu pixel pada batas daerah yang akan di-fill dan kemudian dipilih suatu warna untuk fill.

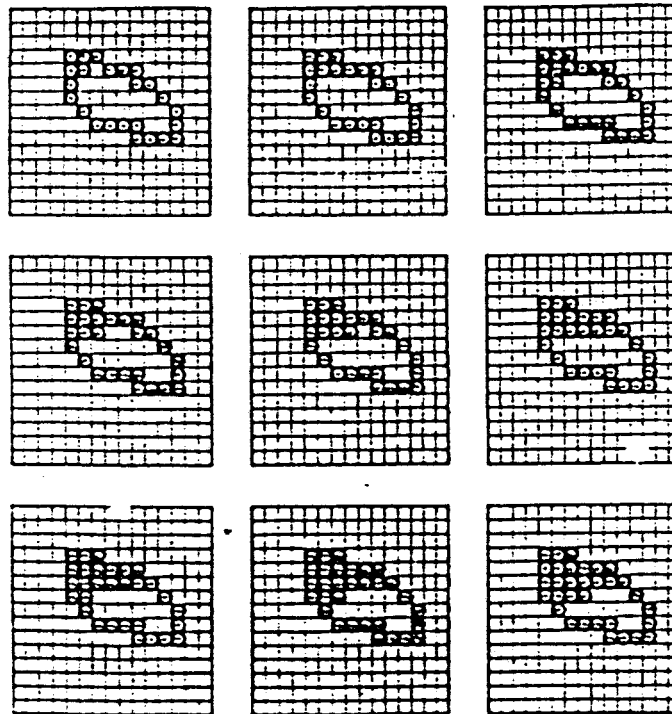
Langkah-langkah dari algoritma ini adalah membaca sebaris daftar obyek. Untuk menggambar titik-titik yang ada dalam obyek tersebut maka digunakan fungsi penggambaran titik yang khusus. Fungsi ini akan menyimpan seluruh kordinat titik di dalam daftar koordinat.

Setelah daftar tersebut diselesaikan, kemudian nilai x dalam y di-sort secara ascending. Kemudian dimulai dari harga y terbesar mulailah mem-plot garis sampai mencapai nilai y terendah.

2.6.10.2. Memuluskan Gambar (Smoothing)

Smoothing bisa dilakukan apabila tampilan memberikan fasilitas warna yang mencukupi, sebagai

gambaran, dengan 16 warna saja tidak akan mungkin didapatkan hasil smoothing yang baik.



GAMBAR 2.15.

PEMILIHAN PIXEL PADA ALGORITMA FLOYD²²⁾

2.6.10.3. Pengecatan (Brushes)

Dalam aplikasi grafik sangatlah dibutuhkan kemampuan pengecatan suatu daerah. Untuk itu diperlukan suatu kwas. Kwas adalah daerah yang ada di tampilan yang bisa digerakkan oleh pemakai. Pada saat dilewati oleh kwas ini maka pixel-pixel akan berubah warna sesuai dengan warna cat yang dipilih.

²²⁾ *Ibid*, hal 92.

2.6.10.4. Palette

Kwas membutuhkan palette. Palette adalah sekelompok warna yang bisa dipilih oleh pemakai.

Dengan memilih warna palette maka pemakai bisa mengecat suatu daerah hanya dengan menggerak-gerakkan kwas.

2.6.11. Penyilangan (Crosshatching)

Jika suatu sistem grafik menggunakan barisan database, yang berisi daftar koordinat (x, y) maka akan lebih mudah untuk mengisi suatu daerah dengan warna. Dengan selektif, pengisian daerah bisa lebih diperkirakan daripada dengan metoda daerah batas pixel. Selain itu daerah yang terputus bisa lebih dikontrol.

Penyilangan (crosshatching) memiliki algoritma yang hampir sama dengan algoritma fill, namun pada penyilangan ini menggunakan interval bukan lagi daerah-daerah batas.

Algoritma ini melakukan penggambaran garis horizontal. Karena elemen penyilangan adalah garis maka pola garis bisa diubah-ubah untuk mendapatkan variasi. Keuntungan algoritma ini adalah sepanjang perbatasan yang dihubungkan akan diperoleh beberapa hole. Daerah yang akan di-crosshatch tidak harus berupa segiempat, tapi bisa juga dalam bentuk segmen-segmen garis yang saling berhubungan dan membentuk daerah.

2.6.12. Format File Gambar

Perangkat-perangkat lunak grafik pada dasarnya mengolah file gambar dalam format-format yang telah ditentukan sebelumnya. Format yang biasa dipakai pada dasarnya bisa dibedakan atas 2 jenis, yaitu:

- a. Berdasarkan vektor (garis)
- b. Berdasarkan titik (bit mapped)

Untuk file gambar yang berdasarkan vektor, maka algoritma yang dipakai lebih kompleks namun fasilitas yang diberikan akan lebih banyak. Fasilitas-fasilitas tersebut dimungkinkan karena setiap kali pengeditan maka akan dilakukan penyimpanan parameter-parameter gambar pada database.

Sedangkan untuk file gambar yang berdasarkan titik, di dalam penyimpanannya tidak membutuhkan algoritma yang sulit. Tetapi untuk melakukan trik-trik tertentu dibutuhkan algoritma yang sangat kompleks.

2.6.13. Pemotongan Gambar (Clipping)

Di dalam aplikasi grafik biasanya ditentukan suatu daerah yang boleh dilakukan pengeditan di dalamnya. Daerah tersebut biasa disebut window.

Jadi, di luar window seluruh pengeditan tidak akan bisa dilakukan. Untuk menentukan window ini bisa dilakukan dengan menentukan empat titik sudut dari window tersebut.

2.6.13.1. Clipping Window

Bila dilakukan penggambaran yang ternyata sampai melebihi batas-batas window yang telah ditentukan maka gambar tersebut akan dipotong (clipping). Banyak metoda pemotongan yang bisa dilakukan.

Suatu gambar yang melebihi batas window bisa dipindahkan ke bagian lain dari layar, namun bisa juga gambar tersebut terpotong (potongan gambar yang lain tidak ditampilkan lagi di layar).

2.7. Peralatan Input Bagi Komputer Grafik

Untuk membuat gambar grafik pada suatu perangkat lunak grafik pada dasarnya diperlukan informasi-informasi. Informasi ini bisa diperoleh dari file atau langsung di-input-kan oleh pemakai.

Dalam usaha mendapatkan hasil yang memuaskan maka dibutuhkan peralatan input yang efektif dan interaktif dengan pemakai. Dalam memasukkan informasi ini biasa dikenal istilah digitizing. Digitizing adalah proses memasukkan data ke dalam data base dari komputer grafik.

Kesulitan berkisar pada bagaimana caranya pemakai dapat meletakkan kursor dan membuat titik pada obyek yang ada di layar. Peralatan yang membantu menyelesaikan masalah di atas itulah yang dinamakan peralatan input untuk grafik.

Peralatan ini mengubah posisi pada alam nyata dan memindahkannya ke dalam format yang bisa digunakan

oleh komputer. Beberapa jenis peralatan input untuk grafik yang biasa dipakai:

- Keyboard

Keyboard adalah peralatan input yang paling sering dipakai. Pada umumnya tombol yang dipakai adalah tombol anak panah. Tombol anak panah ini digunakan untuk mengatur gerakan kursor yang berdasarkan langkah-langkah dengan interval tertentu.

- Potensiometer

Satu atau lebih knob pemutar bisa dipakai untuk menempatkan kursor dan mengatur fungsi-fungsi grafik lain. Biasanya dipakai 2 knob yaitu satu untuk posisi x dan satu posisi y. Knob-knob ini terdiri dari potensiometer.

Dengan potensiometer maka gerakan kursor tidak lagi dibatasi oleh langkah-langkah dengan interval tertentu seperti yang terjadi pada keyboard, namun gerakan kursor menjadi lebih bebas.

- Joystick

Pada dasarnya prinsip kerja dari joystick ini sama dengan potensiometer, karena joystick ini juga terbuat dari potensiometer yang bisa berputar dalam 2 dimensi

- Meja grafik

Sementara orang menginginkan agar cara menggambar pada komputer semudah cara menggambar secara tradisional yaitu memakai pensil dan kertas. Dari keinginan inilah dibuatlah suatu meja grafik.

Meja grafik ini berbentuk suatu permukaan lempengan yang besar di mana pemakai hanya perlu menggerakkan stylus untuk menempatkan kursor pada posisi yang diinginkan. Stylus ini memiliki pena ballpoint pada ujungnya, sehingga sekilas seakan-akan menggambar dengan stylus di atas kertas yang ada pada lempengan.

2.7.1. Konversi Data Input

Ada 2 metoda yang bisa dipakai untuk mengubah posisi pedal potensiometer menjadi suatu informasi yang bisa dipakai oleh komputer, yaitu:

- a. Analog to Digital Converter
- b. Timer

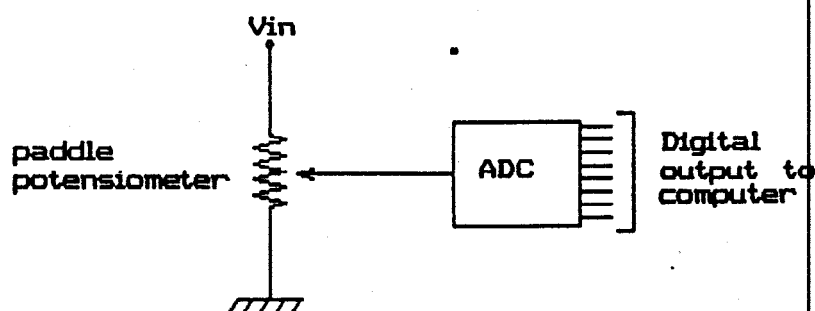
2.7.1.1. Metoda Analog to Digital Converter

Di dalam metoda A/D converter, penggunaan pedal potensiometer adalah sebagai variabel resistor dalam pembagi tegangan. Seperti yang terlihat pada gambar 2.16.

Pedal potensiometer merupakan sumber tegangan yang variabel yang akan diteruskan ke rangkaian A/D converter yang sebenarnya. Rangkaian A/D converter inilah yang akan mengubah tegangan ke dalam angka digital.

Output dari A/D converter ini mudah dibaca oleh perangkat lunak melalui port yang telah ditentukan, misalnya untuk IBM PC adalah port dengan alamat 201

Hexa. Kelebihan dari A/D converter adalah secara logika output yang dihasilkan dapat dengan mudah langsung digunakan oleh program grafik. Selain itu dengan metoda ini, nilai yang diperoleh tidak dipengaruhi oleh besaran resistif atau kapasitif dari kabel yang terhubung.



GAMBAR 2.16.

METODA ANALOG TO DIGITAL CONVERTER²³⁾

Namun chip A/D converter ini biasanya memiliki harga yang mahal sehingga orang lebih suka memilih metoda yang lain yaitu metoda timer.

2.7.1.2. Metoda Timer

Pada metoda timer ini pedal potensiometer bertindak sebagai bagian resistif dari rangkaian pengatur RC timer.

Timer secara berselang ditrigger melalui perangkat lunak, dan periode yang dilewati selama timer on adalah sebanding dengan konstanta waktu RC yang pada dasarnya

²³⁾ Artwick, *op cit*, hal 127.

adalah posisi pedal potensiometer.

Mikroprosesor akan menghitung nilai awal pada saat timer ditrigger dan nilai akhir pada saat time out. Nilai ini digunakan oleh program sebagai posisi dari pedal

Keuntungan dari metoda timer ini adalah harga rangkaian yang relatif lebih murah, namun metoda ini kurang dapat menghasilkan unjuk kerja sistem grafik yang tinggi, karena banyak pengaruh yang timbul dan menyebabkan posisi pedal yang didapatkan oleh perangkat lunak tidak asli lagi. Pengaruh itu antara lain adalah harga resistif atau kapasitif dari kabel yang menghubungkan pedal potensiometer dengan rangkaian timer. Bertambah panjang kabelnya maka bertambah besar pula nilai resistifnya sehingga bertambah besar pengaruhnya pada konstanta waktu RC.

2.8. Analog to Digital Converter

Analog to Digital Converter (A/D Converter atau ADC) berfungsi untuk mengkonversikan sinyal analog ke dalam bentuk kombinasi output biner (digital) yang sesuai. Cukup banyak metoda yang dapat digunakan untuk menghasilkan rangkaian A/D converter. Pemakaian metoda-metoda tersebut tergantung pada faktor-faktor mana yang perlu diandalkan. Faktor-faktor yang perlu diperhitungkan antara lain adalah kecepatan konversi (conversion time), ketelitian (accuracy), kestabilan

(stability) maupun faktor biaya (cost). Untuk membangun ADC ini secara garis besar ada 2 macam metoda yang dapat digunakan yaitu metoda open-loop dan metoda closed-loop (feedback).

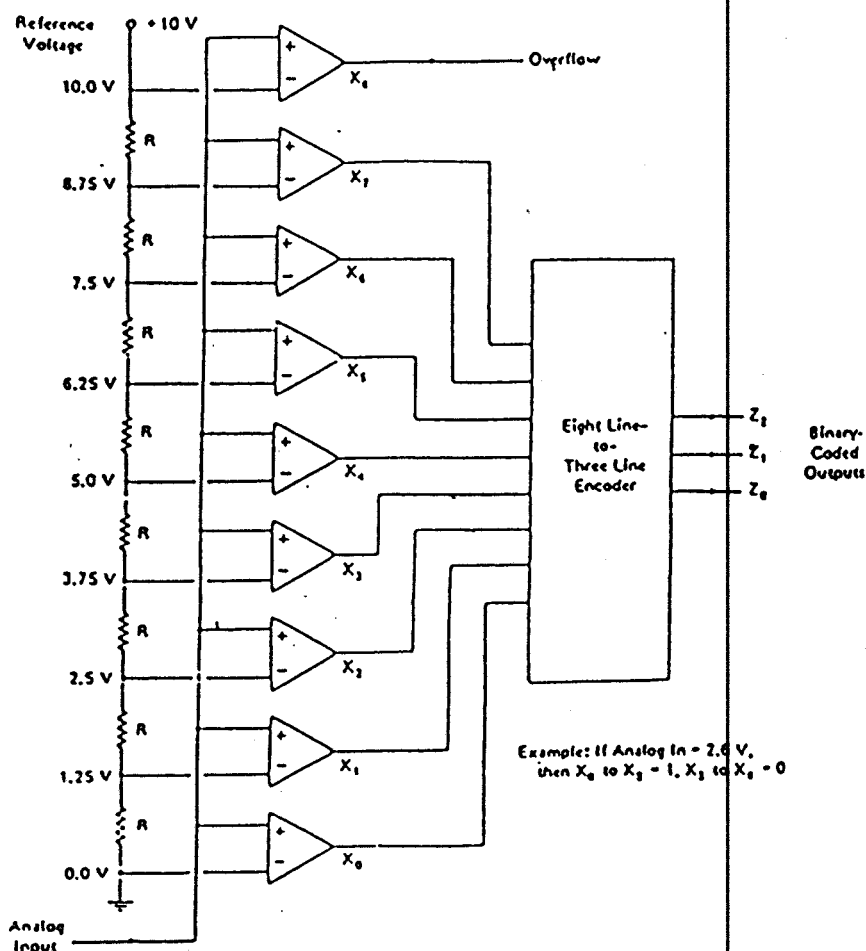
Dalam metoda open-loop kita mengenal Flash ADC, Time Window ADC, Slope Converter, Dual Slope Converter (banyak digunakan pada alat-alat ukur). Sedangkan pada metoda closed-loop (dengan feedback) kita mengenal tipe-tipe ADC seperti Single Counter, Tracking ADC, Successive Approximation ADC.

Dalam pembahasan ini hanya akan dijelaskan Flash ADC dan Successive Approximation ADC saja untuk melihat perbandingan antara kedua tipe ADC tersebut, keuntungan, kerugian dan pertimbangan-pertimbangan apakah yang harus ditentukan dalam memilih ADC.

2.8.1. Flash ADC

Rangkaian ADC ini merupakan yang paling sederhana tetapi sekaligus merupakan ADC yang tercepat waktu konversinya. Untuk ADC n bit maka dibutuhkan komparator sebanyak $2^n - 1$ buah.

Input tegangan analog yang akan diubah ke digital dipakai sebagai input bagi semua komparator, sedangkan input yang lain dari semua komparator didapatkan dari tegangan referensi yang dibagi-bagi oleh sederetan resistor, tegangan ini disebut tegangan batas. Jika tegangan input melebihi dari tegangan batas maka output komparator akan high. Tegangan referensi pada komparator



GAMBAR 2.17.

FLASH ADC

teratas merupakan tegangan full-scale yang diinginkan. Jika tegangan input melebihi tegangan tersebut maka ADC akan mengeluarkan sinyal overflow dan semua komparator akan high. Output komparator kemudian dimasukkan ke rangkaian Encoder untuk diterjemahkan menjadi binary

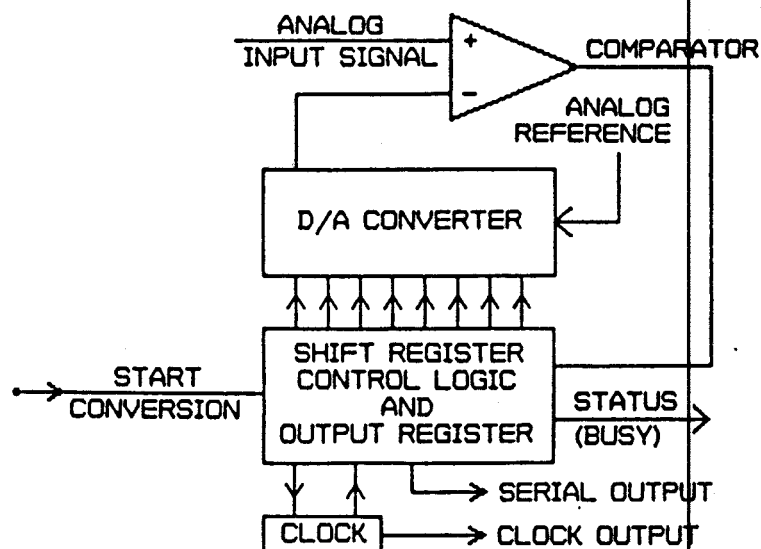
code yang sesuai. Jadi waktu konversinya ditentukan oleh waktu switching dari komparator dan waktu delay dari rangkaian Encoder.

Walaupun ADC ini bekerjanya sangat cepat tetapi kurang efisien untuk jumlah bit yang besar, sebab dibutuhkan banyak sekali komparator. Demikian juga output dari komparator yang belum berbentuk biner sehingga masih dibutuhkan satu rangkaian lagi yaitu Encoder. Dengan demikian untuk jumlah bit yang besar tidak hanya menaikkan biaya namun juga memerlukan stage tambahan yang otomatis akan memperbesar waktu konversinya juga.

2.8.2. Successive Approximation ADC

ADC ini merupakan tipe closed-loop yang terbaik dan tercepat karena kecepatan konversinya selalu tetap tidak tergantung pada besarnya sinyal input analog. Di samping itu ADC jenis ini dapat dibuat dengan jumlah bit yang banyak (resolusi tinggi). ADC ini dibentuk dari beberapa komponen yaitu DAC (Digital to Analog Converter), komparator dan beberapa register serta rangkaian kontrol.

Proses konversi dimulai dengan memberikan sinyal start konversi. Dengan sinyal tersebut mengakibatkan MSB (Most Significant Bit) pada input DAC berharga '1' sedangkan bit yang lain berharga '0'. Output DAC yang terjadi ($1/2$ Full Scale) dibandingkan dengan input analog yang akan dikonversikan.



GAMBAR 2.18.

DIAGRAM BLOK SUCCESSIVE APPROXIMATION ADC²⁴⁾

Dari output komparator dapat diketahui mana yang lebih besar. Bila tegangan output DAC lebih kecil dari tegangan input maka bit MSB tetap '1', sedangkan bila output DAC lebih besar maka MSB berubah menjadi '0'. Kemudian bit berikutnya (1 bit lebih rendah dari MSB) dibuat = '1' dengan keadaan MSB tetap seperti setelah proses perbandingan pertama tadi. Dari keadaan input digital ini akan dihasilkan output analog yang juga akan dibandingkan lagi dengan tegangan input untuk menentukan apakah keadaan bit kedua akan diubah atau tidak. Proses di atas berlangsung sampai LSB (Least Significant Bit), sehingga akan diperoleh output digital dari input DAC yang terakhir. Proses penggeseran bit dari MSB sampai

24) Sheingold, Daniel H., Analog-Digital Conversion Notes, Analog Device Inc, 1977, hal 122.

LSB dilakukan oleh shift register yang waktunya diatur oleh pulsa clock. Jika proses konversi telah selesai maka output status bit akan mengeluarkan sinyal 'End Of Conversion' yang menyatakan bahwa output digital ADC dapat diambil sebagai besaran digital yang mewakili input analog yang telah dikonversi.

BAB III

PERENCANAAN DAN PEMBUATAN PERANGKAT KERAS

Dalam bab ini akan dibahas tahap-tahap perencanaan dan pembuatan perangkat keras. Alat (perangkat keras) yang dibuat berupa interface yang nantinya digunakan untuk berhubungan dengan komputer IBM PC-XT. Interface ini bersama-sama dengan papan grafik yang sudah ada akan merupakan peralatan input bagi program grafik yang akan diterangkan pada Bab 4.

Fungsi papan grafik ini sebenarnya adalah untuk menggerakkan kursor. Gerakan ini bisa dilakukan karena pada meja gambar ini terdapat 2 buah lengan bantu. Pada tiap lengan dipasangkan potensiometer. Pada masing-masing potensiometer diberikan tegangan. Sehingga, perubahan posisi lengan yang menyebabkan perubahan harga resistan dari potensiometer, juga akan menyebabkan perubahan tegangan outputnya. Output ini akan diberikan kepada perangkat keras yang dibuat untuk kemudian dikonversikan ke dalam bentuk digital. Informasi digital ini akan diteruskan kepada komputer IBM PC-XT. Tiap data

digital akan mewakili posisi tiap potensiometer.

Potensiometer yang satu menunjukkan posisi dari x sedang yang lainnya menunjukkan posisi y.

Masalah yang timbul sebenarnya adalah bagaimana caranya menghubungkan meja gambar ini dengan komputer IBM PC-XT.

Hubungan yang dimaksudkan di sini adalah memindahkan informasi baik dari atau ke komputer dengan interface yang dibuat.

Untuk itu bisa digunakan prinsip komputer kontrol. Prinsip tersebut adalah komputer harus mempunyai cara supaya bisa mengetahui respon dari perangkat keras eksternal terhadap kontrol yang diberikan. Oleh sebab itu komputer harus mampu memonitor peralatan eksternal tersebut.

Informasi atau respon yang diterima oleh komputer dapat diolah dan dikirim kembali ke peralatan eksternal.

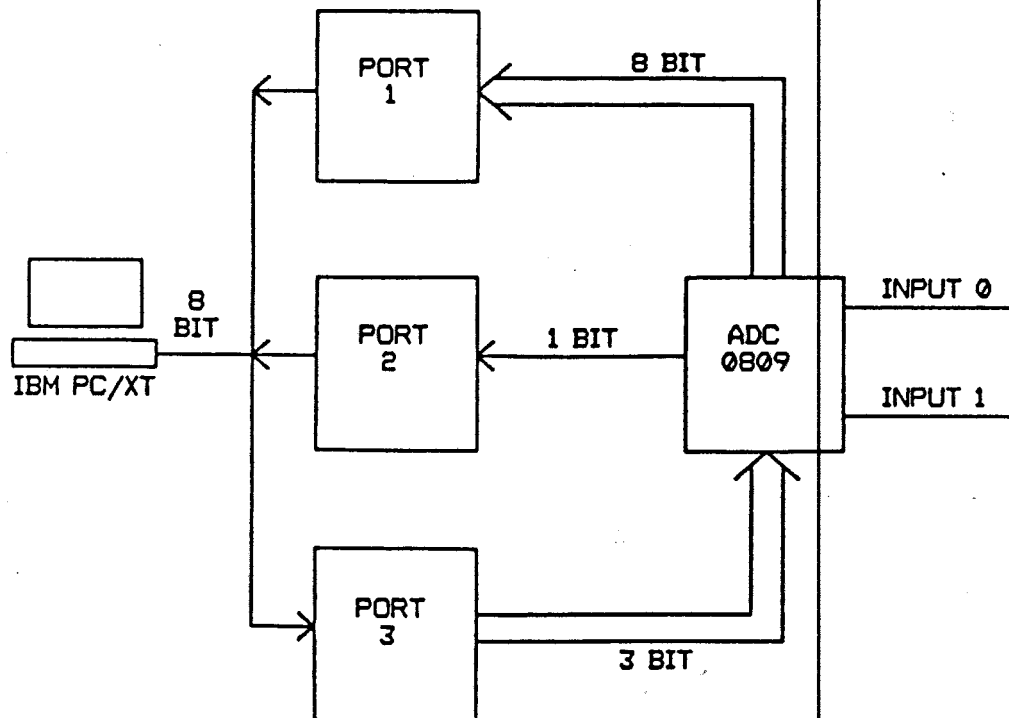
Dengan demikian ada 2 proses utama pada komputer kontrol, yaitu:

- Komputer menerima informasi dari perangkat keras eksternal.
- Komputer mengirimkan informasi ke perangkat keras eksternal.

3.1. Perangkat Keras Untuk IBM PC-XT

Pada bagian ini akan dibahas mengenai perangkat keras eksternal yang akan dibuat untuk mengeluarkan atau menerima data dari IBM PC-XT.

Diagram blok untuk perangkat keras yang akan dibuat bisa dilihat pada gambar berikut :



GAMBAR 3.1.

DIAGRAM BLOK PERANGKAT KERAS

Dari diagram blok di atas bisa dilihat ada 3 port yang akan digunakan. Namun 3 port tersebut hanya memiliki 2 alamat, dengan rincian sebagai berikut:

- Port 1 menempati 1 alamat (alamat 696 atau heksa 2B8) berfungsi sebagai port input. Port 1 ini bertugas untuk menampung data 8 bit yang merupakan hasil konversi sinyal analog yang dilakukan oleh ADC 0809.
- Port 2 dan port 3 menempati 1 alamat (alamat 697 atau heksa 2B9) :

* Port 2 berfungsi sebagai port input

Dari port 2 ini IBM PC-XT dapat memonitor sinyal EOC

(End of Conversion) yang dihasilkan oleh ADC 0809.

* Port 3 berfungsi sebagai port output

Port 3 ini akan digunakan oleh IBM PC-XT untuk mengirimkan/memberikan sinyal SC (Start of Conversion), alamat dari input analog yang akan diaktifkan serta OE (Output Enable).

Bagian lain yang tidak kalah penting adalah ADC 0809. Bagian inilah yang akan mengubah sinyal analog yang dihasilkan oleh potensiometer yang ada pada masing-masing lengan bantu menjadi data digital 8 bit yang akan dikirimkan ke IBM PC-XT melalui port di atas.

Data 8 bit ini merupakan sumber informasi utama bagi perangkat lunak untuk kemudian dipresentasikan pada layar monitor.

Dua buah data 8 bit yang didapatkan dari kedua potensiometer akan mewakili posisi sebuah titik yang ada pada papan grafik.

3.2. Rangkaian Output Untuk IBM PC-XT

Di sini terdapat 4 bagian utama yang digunakan untuk mengeluarkan data dari komputer IBM PC-XT, yaitu:

1. Rangkaian enable output untuk peralatan eksternal.
2. Output Write Strobe (-IOW) dari komputer.
3. Write Strobe untuk rangkaian.
4. Output Latch.

Empat bagian ini bekerja bersama-sama untuk

melakukan fungsi output.

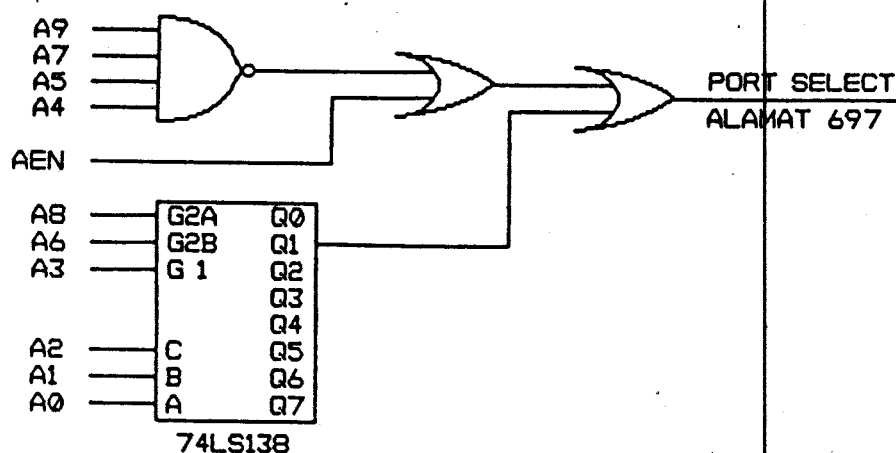
3.2.1. Rangkaian Enable Output

Akan dibahas fungsi dari rangkaian enable dalam operasi output. Pada saat rangkaian enable aktif maka berarti komputer telah menunjuk alamat rangkaian eksternal yang dituju. Seperti yang dijelaskan pada bab 2, IBM PC-XT dapat berkomunikasi dengan lebih dari 64.000 rangkaian eksternal yang berbeda. Untuk membedakan rangkaian eksternal mana yang dipilih, maka harus dilengkapi dengan pengkodean sehingga dapat diinformasikan rangkaian eksternal yang dituju. Sinyal enable dihasilkan dari kombinasi logika yang ada pada saluran address bus sistem slot yaitu A0 sampai A15. Kombinasi A0 sampai A15 ini akan membentuk suatu alamat untuk peralatan eksternal (port).

Seperti yang dijelaskan pada bab 2.3. maupun bab 2.3.1. ternyata dari 16 bit yang disediakan prosesor 8088 hanya digunakan 10 bit terendah saja yaitu A0 sampai A9.

Untuk mencari alamat port yang bisa digunakan untuk keperluan peralatan yang akan dibuat bisa dilihat pada gambar 2.7. Perlu diperhatikan bahwa pengalamatan port yang sudah digunakan oleh peralatan eksternal yang lain tidak dapat digunakan lagi. Oleh sebab itu harus dipilih alamat yang tidak digunakan (Not Used). Untuk peralatan yang akan dibuat ini, digunakan alamat 696 dan 697.

Rangkaian enable untuk alamat 697 bisa dilihat pada gambar 3.2.



GAMBAR 3.2.

RANGKAIAN ENABLE

Sinyal yang dihasilkan oleh rangkaian enable biasa disebut Port Select. Untuk dapat menghasilkan lebih dari satu sinyal Port Select maka pada saluran alamat A0 - A2 dimasukkan dalam decoder. Salah satu dari output decoder tadi dimasukkan dalam gerbang or bersama sinyal kombinasi saluran alamat A3 - A9.

Sinyal AEN (Address Enable) menandakan bahwa address bus sistem slot dikendalikan oleh prosessor 8088. Ketika jalur ini berlogika nol, maka pada address bus sistem slot A0 - A15 berisi alamat port yang sah (valid). Bila sinyal AEN ini berlogika satu maka address bus sistem slot bernilai acak dan tidak dapat mengkodekan sebuah alamat port.

Untuk dapat mengeluarkan data maupun untuk

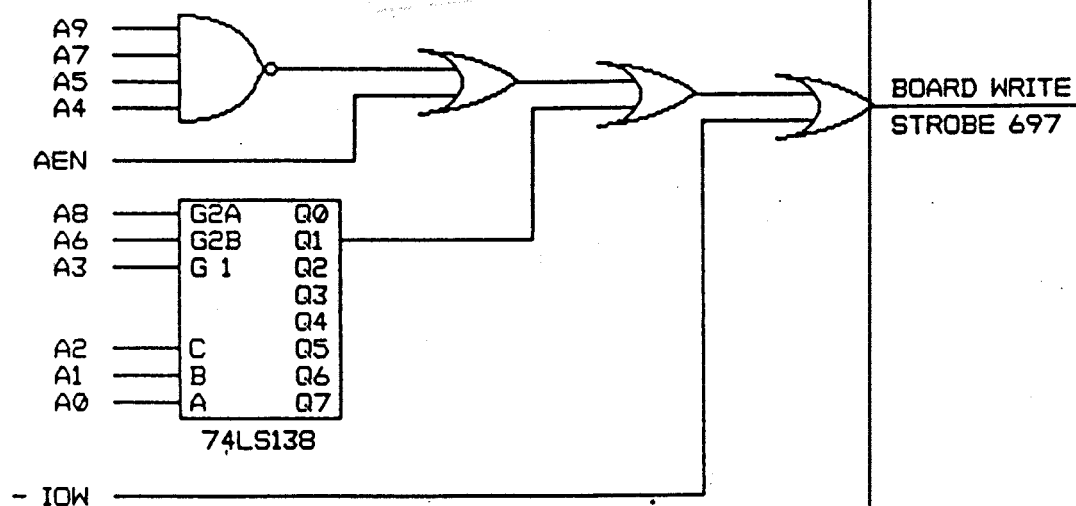
mengambil data dari IBM PC-XT, diperlukan sinyal board select. Sinyal ini didapatkan dari kombinasi antara sinyal port select dan sinyal AEN. Sinyal board select ini selalu berlogika satu dan apabila sinyal port select dan AEN sama-sama berlogika nol maka sinyal board select akan berlogika nol. Berarti board select sedang aktif.

3.2.2. Output Write Strobe

Sinyal -IOW disebut output write strobe karena sinyal ini hanya aktif pada saat instruksi OUT dilaksanakan. Selama instruksi OUT, sinyal ini akan berlogika nol dan data bus berisi data yang valid. Pada saat tersebut data akan diterima oleh rangkaian eksternal yang alamatnya telah ditentukan sebelumnya, instruksi ini selalu menyertakan alamat port output dalam pelaksanaannya.

3.2.3. Board Write Strobe

Kombinasi sinyal output write strobe dengan sinyal board select menghasilkan sinyal baru yang disebut sinyal board write strobe. Bila sinyal -IOW dan sinyal board select masing-masing berlogika nol maka sinyal board write strobe akan berlogika nol dan ini menandakan rangkaian eksternal yang terpilih dapat mengambil data dari data bus. Gambar 3.3. memperlihatkan rangkaian lengkap untuk membangkitkan sinyal board write strobe. Sinyal ini akan berlogika nol bila instruksi OUT dilaksanakan.



GAMBAR 3.3.

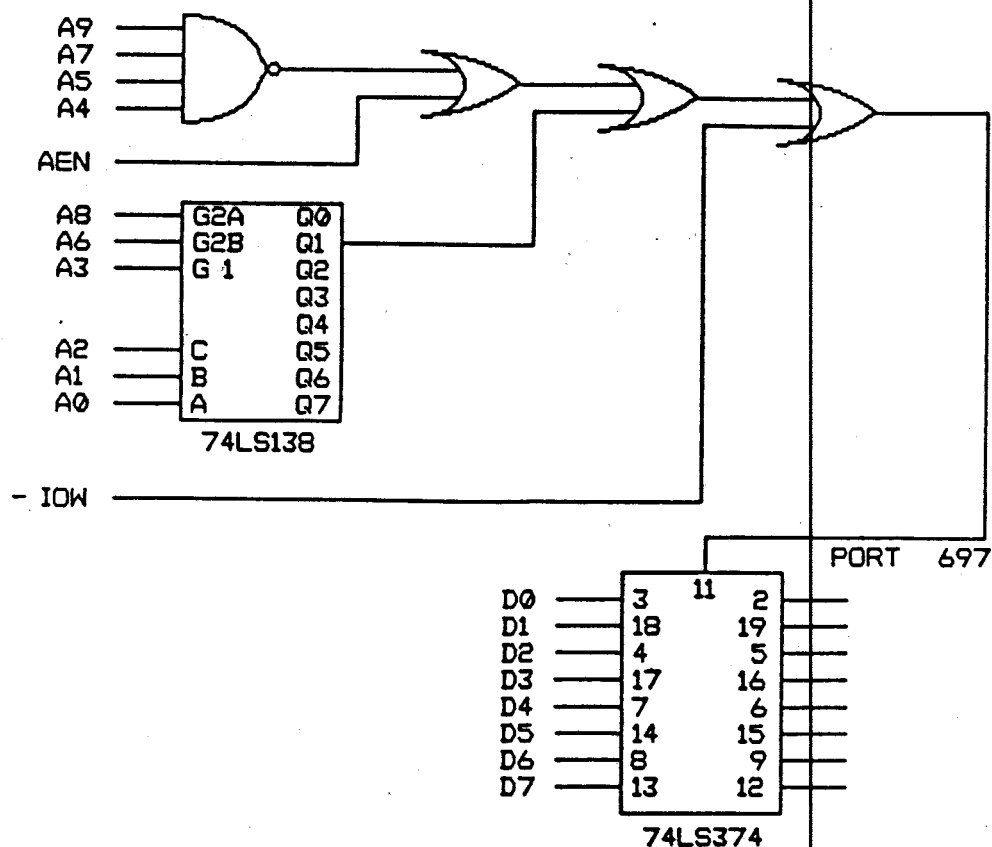
RANGKAIAN LENGKAP BOARD WRITE STROBE

Terlihat pada gambar tersebut bahwa hanya ada sebuah sinyal board write strobe, karena seperti yang telah dijelaskan pada bagian awal dari bab ini, untuk alat yang dibuat hanya dibutuhkan sebuah port output yang fungsinya untuk menerima sinyal-sinyal dari IBM PC-XT yang nantinya akan bertindak sebagai sinyal SC (Start of Conversion), sinyal alamat dari input analog yang dipilih, sinyal OE (Output Enable).

3.2.4. Output Latch

Sebenarnya dengan sinyal board write strobe sudah didapatkan data yang sah pada data bus, tetapi data yang didapat lewat data bus lamanya tidak lebih dari 840 nanodetik. Karena pendeknya waktu pengiriman data lewat data bus maka data perlu di-latch terlebih dahulu. Latch ini mempunyai fungsi sangat penting untuk menahan data

sementara sehingga data dapat dikirim ke peralatan eksternal. Input dari latch ini dihubungkan dengan data bus (lihat gambar 3.4.). Pada saat yang sama sinyal board write strobe memberikan pulsa ke latch. Sesudah data tertulis ke dalam latch, data ini akan terus tertahan atau tersimpan sampai komputer melaksanakan instruksi OUT yang berikutnya pada alamat port output yang sama.



GAMBAR 3.4.

RANGKAIAN LENGKAP PORT OUTPUT PADA ALAMAT 697

Gambar 3.4. memperlihatkan rangkaian lengkap untuk port output dengan alamat 697. Pada rangkaian ini

digunakan latch 8 bit dengan type 74LS374. Pin nomor 11 dari latch ini digunakan untuk strobe input. Sinyal ini aktif bila sinyal ini bergerak dari logika nol ke logika satu. Pada saat sinyal ini aktif maka data yang ada pada input akan ditangkap dan disimpan dalam latch.

3.3. Rangkaian Input untuk IBM PC-XT

Seperti pada rangkaian output maka pada rangkaian input ini juga dibagi menjadi empat bagian :

1. Rangkaian Enable Input
2. Output Read Strobe (-IOR) dari komputer
- 3 Read Strobe untuk rangkaian
4. Buffer Input

3.3.1. Rangkaian Enable Input

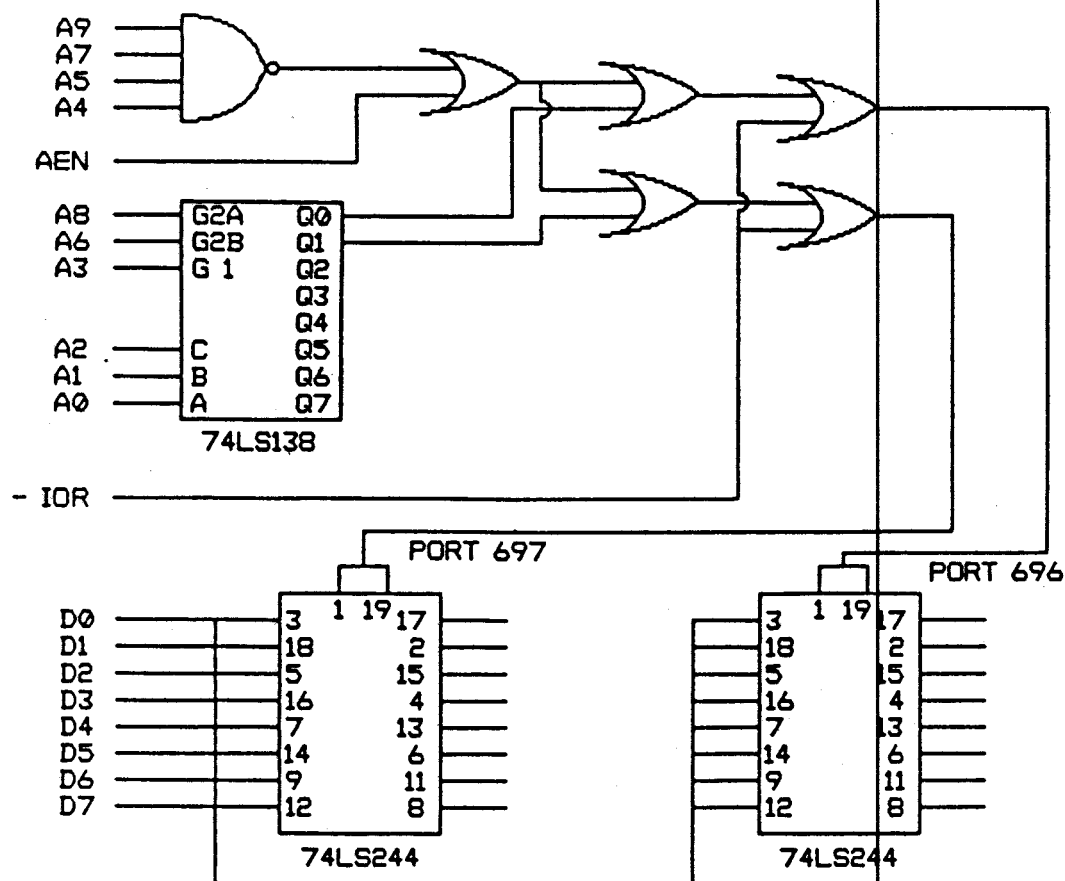
Rangkaian enable input ini sama seperti pada rangkaian enable output, hanya saja alamat untuk portnya berbeda. Rangkaian enable ini menjadi lebih hemat karena ada satu port yang memiliki 2 fungsi yaitu input maupun output.

3.3.2. Output Read Strobe

Sinyal -IOR disebut output read strobe karena sinyal ini aktif pada saat instruksi IN dilaksanakan. Selama pelaksanaan instruksi IN, sinyal ini akan berlogika nol dan data bus berisikan data yang valid. Pada saat inilah data akan dikirim dari rangkaian eksternal yang alamatnya telah ditentukan sebelumnya

rangkaian eksternal tetapi untuk dapat menggunakan data bus sistem slot yang sama diperlukan rangkaian buffer yang mempunyai output tiga keadaan (tri state buffer), dimana keadaan output buffer dapat dikontrol melalui sinyal board read strobe.

Gambar 3.6. memperlihatkan rangkaian lengkap yang mampu menerima data dari rangkaian eksternal dengan alamat 696 dan 697.



GAMBAR 3.6.

RANGKAIAN LENGKAP PORT INPUT PADA ALAMAT 696 & 697

Pada saat sinyal board read strobe aktif atau berlogika nol menyebabkan pin nomor 1 dari IC 74LS244

3.4. Rangkaian ADC

Selain port-port yang telah dijelaskan di atas, maka ada satu bagian lagi yang tak kalah penting. Bagian ini adalah bagian pengkonversi sinyal. Sinyal yang akan dikonversi di sini adalah sinyal analog yang diperoleh dari potensiometer yang ada pada lengan bantu papan grafik. Sinyal analog ini harganya bervariasi, namun maksimal berharga +5 Volt. Karena pada papan grafik terdapat 2 buah potensiometer berarti akan ada 2 sinyal analog yang akan dikonversikan. Sinyal analog ini akan dikonversikan ke dalam data digital 8 bit oleh rangkaian Analog to Digital Converter (ADC).

Dalam pembuatan perangkat keras ini akan digunakan ADC 0809. ADC ini bertipe SAR (Successive Approximation ADC) yang mempunyai waktu konversi cukup singkat yaitu 100 mikro detik dan dapat di-interface ke semua jenis mikroprosesor. Di samping itu ADC tipe ini banyak dijual di pasaran serta harganya relatif murah.

Tegangan supply untuk rangkaian ADC ini diambilkan dari IBM PC-XT yang sudah tersedia pada slot. Demikian juga tegangan referensi positif (Ref +) dihubungkan dengan tegangan +5 Volt yang ada pada slot. Untuk ground dari ADC (GND) maupun tegangan referensi negatif (Ref -) dihubungkan dengan ground yang ada pada slot. Demikian juga untuk tegangan supply maupun ground bagi chip-chip lainnya seluruhnya diambilkan dari slot.

ADC 0809 mampu menangani 8 buah input analog yang

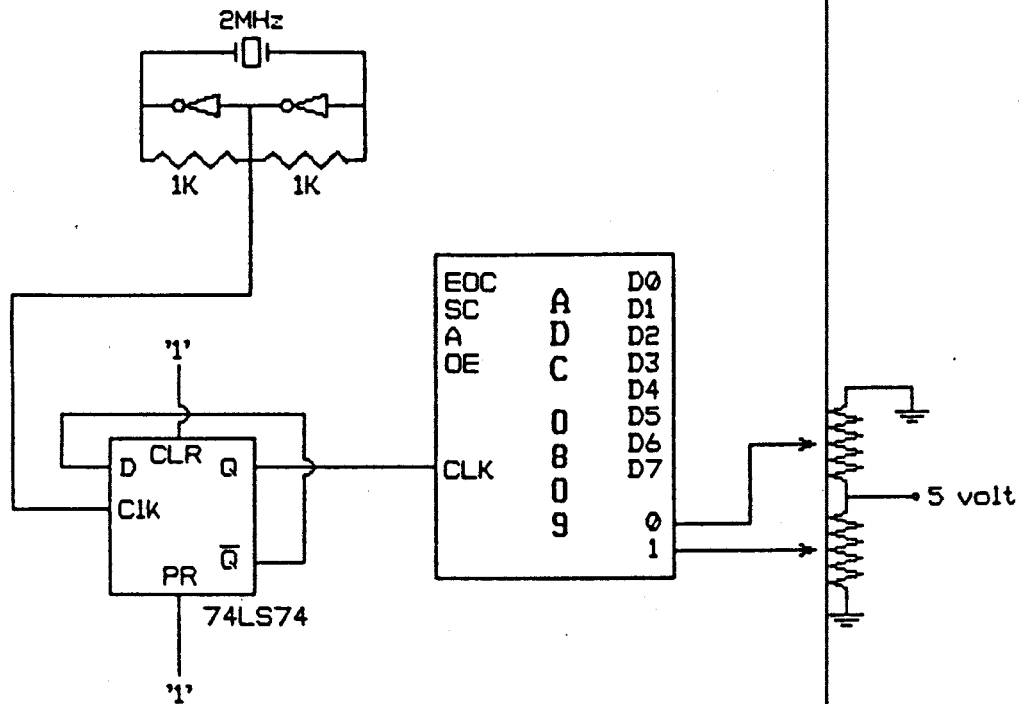
di-multiplex. Untuk memilih satu di antara kedelapan input analog tersebut bisa dilakukan dengan membuat kombinasi pada pin address input yaitu A, B, C (pin 25, 24, 23). Karena pada perangkat keras yang akan dibuat ini hanya ada 2 input analog maka hanya diperlukan kombinasi logika pada address input A sedangkan address input yang lain (B dan C) di-ground.

Seperti dijelaskan di atas, ADC 0809 mampu menangani 8 buah input analog. Input-input analog tersebut diletakkan pada pin 26, 27, 28, 1, 2, 3, 4 dan 5. Namun di sini hanya akan digunakan 2 input saja yang akan dihubungkan pada pin 26 dan 27, sedangkan pin-pin input analog yang lain akan dihubungkan dengan ground.

Untuk mengkonversikan sinyal analog, ADC 0809 ini juga membutuhkan sinyal clock. Sinyal clock ini dimasukkan melalui pin 10. Berdasarkan buku data maka diketahui bahwa ADC 0809 memerlukan clock yang besarnya berkisar antara 123 sampai 1234 KHz. Pada perencanaan perangkat keras ini akan digunakan clock sebesar 1000 KHz.

Untuk mendapatkan clock sebesar 1000 KHz digunakan kristal 2000 KHz. Dari rangkaian clock yang bisa dilihat pada gambar 3.8. akan didapatkan clock sebesar 2000 KHz. Sinyal ini akan dibagi 2 oleh rangkaian divider (74LS74). Dengan demikian akan diperoleh sinyal clock sebesar 1000 KHz dari pin 5 chip 74LS74.

Proses konversi dimulai dengan input pulsa ALE

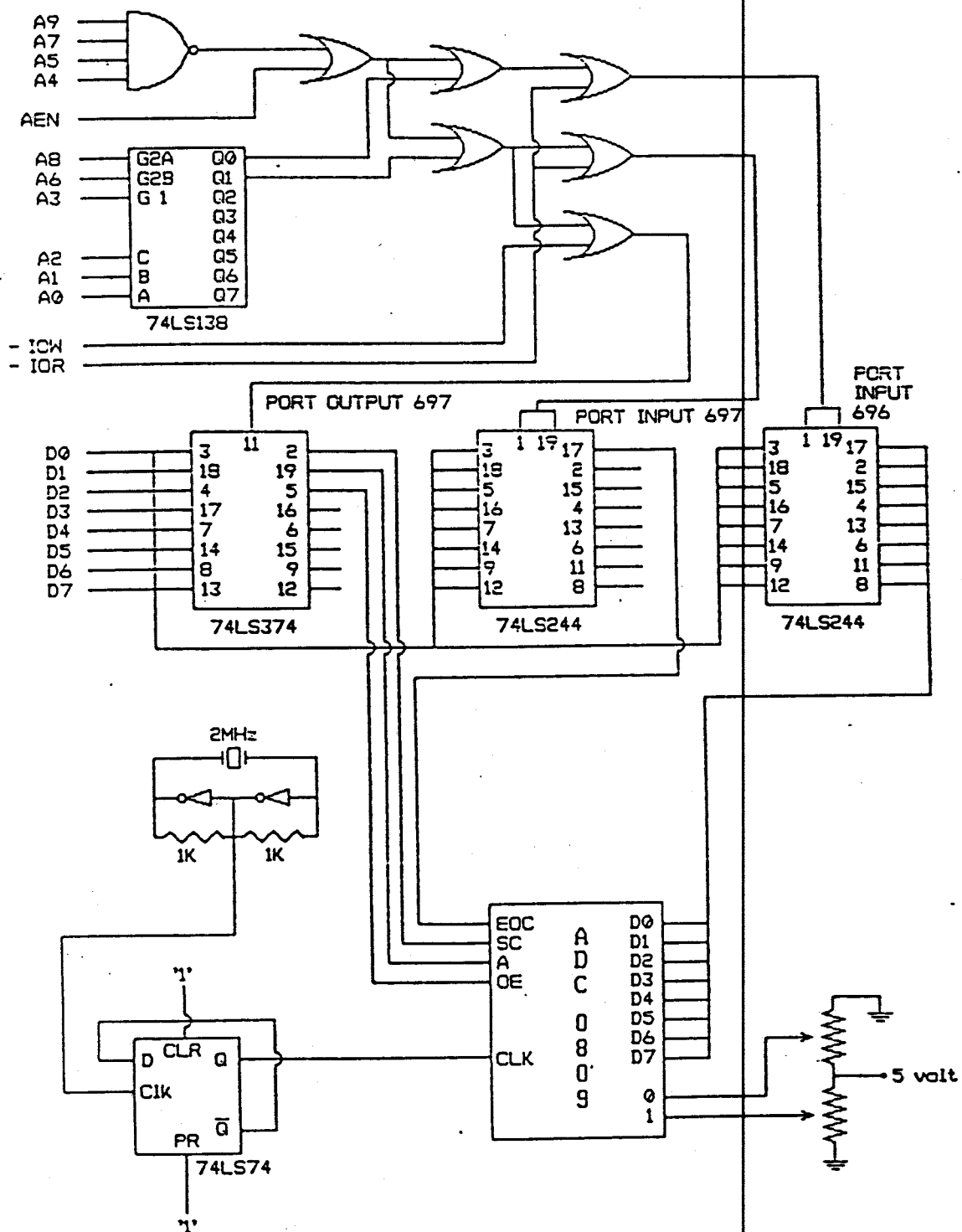


GAMBAR 3.8.

RANGKAIAN ADC

(Address Latch Enable) tinggi. Kemudian kombinasi address A, B & C dimasukkan dalam decoder untuk memilih input analog yang akan dikonversi serta sekaligus input analognya dihubungkan dengan bagian analog multiplexer. Bersamaan dengan itu sinyal SC (Start of Conversion) harus diberikan. Sinyal SC ini merupakan pulsa positif edge (dari '0' menuju '1' dan kemudian '0' lagi). Dengan adanya sinyal SC ini maka latch output akan reset. Konversi akan segera dimulai pada saat pulsa SC menuju '0'. Suatu proses konversi akan di-interrupt apabila ada pulsa SC yang baru.

Selama proses konversi, sinyal EOC (End of Conversion) yang dihasilkan akan berlogika '0'. Pada saat konversi selesai maka ADC 0809 akan menghasilkan



GAMBAR 3.9.
RANGKAIAN LENGKAP

sinyal EOC berlogika '1'. Ini berarti data 8 bit hasil konversi sudah ada di dalam output latch.

Untuk dapat mengeluarkan data hasil konversi tersebut maka dibutuhkan sebuah sinyal lagi, yaitu sinyal OE (Output Enable). Sinyal OE harus diberi harga '1' untuk dapat memerintahkan latch output agar mengeluarkan 8 bit data hasil konversi.

Pada perangkat yang direncanakan ini pemberian maupun pemantauan sinyal dari/ke ADC dilakukan melalui port-port yang telah diterangkan sebelumnya. Proses pengiriman atau pengambilan sinyal tersebut dilakukan oleh perangkat lunak yang akan dijelaskan pada bab 4.

Sinyal untuk SC (Start of Conversion) ditempatkan pada bit ke-0 dari port output 697. Address A ditempatkan pada bit ke-1 dari port output 697. Sedangkan sinyal untuk OE ditempatkan pada bit ke-2 dari port output 697.

Pemantauan sinyal EOC dapat dilakukan dengan membaca bit ke-0 dari port input 697.

Dan yang terakhir, data hasil konversi bisa dibaca pada bit 0 sampai 7 dari port input 696.

BAB IV

PERENCANAAN DAN PEMBUATAN PERANGKAT LUNAK

Di dalam merencanakan perangkat lunak yang akan dibuat, maka langkah awal yang harus dilakukan adalah merancang sistem grafik yang akan digunakan. Sistem grafik ini mempunyai banyak parameter, yaitu mulai dari struktur program, mode grafik yang diinginkan, format file gambar yang akan digunakan dan lain lain.

4.1. Struktur Program

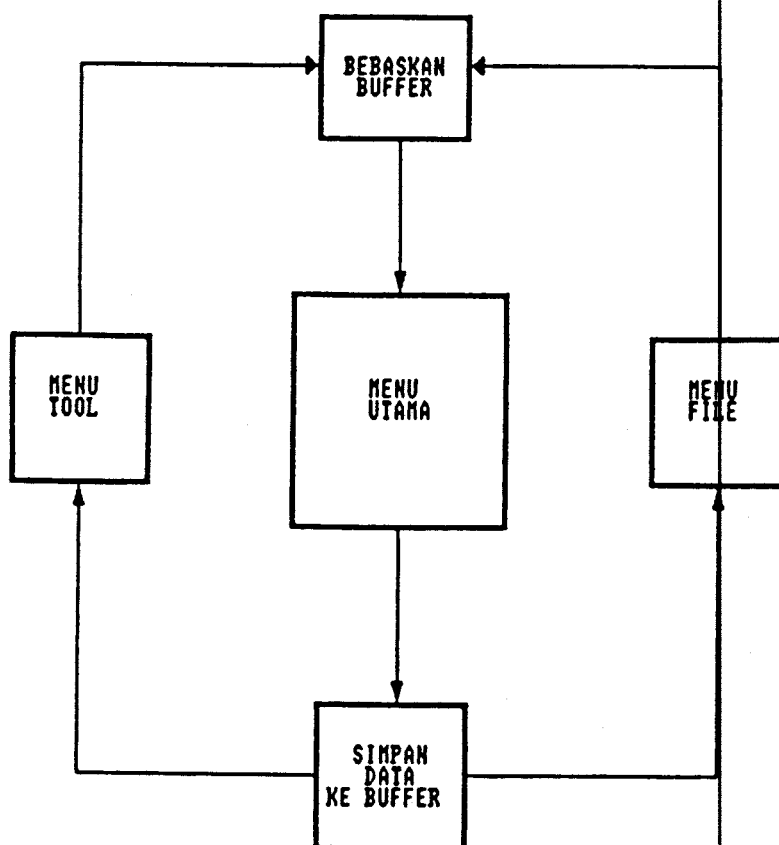
Prinsip keberhasilan program grafik tidaklah berbeda dengan prinsip keberhasilan program pada umumnya.

Langkah awal adalah membagi segala masalah di dalam pemrograman menjadi bagian-bagian yang lebih kecil sehingga akan lebih mudah untuk diselesaikan.

Langkah kedua adalah membuat program yang mempunyai struktur yang jelas. Program komputer biasanya dibuat dari masalah yang umum ke masalah yang lebih spesifik.

Prinsip pertama dan kedua ini sebenarnya mempunyai hubungan yang sangat erat dan tidak bisa dipisahkan.

Dengan mengikuti prinsip-prinsip itu program ini dirancang. Dengan struktur yang jelas membuat program ini mudah untuk dianalisa atau dipelajari.



GAMBAR 4.1.

DIAGRAM BLOK PROGRAM

Struktur program yang direncanakan adalah sebagai berikut. Segala sesuatu diatur oleh menu utama, menu utama ini merupakan menu untuk mengedit gambar maupun untuk memanggil menu-menu yang lain. Untuk pergi ke menu lain maka data gambar harus disimpan dulu pada buffer

yang berupa array dinamik. Baru setelah menu tersebut selesai dijalankan, buffer dibebaskan dan kontrol dikembalikan lagi ke menu utama.

4.2. Manipulasi Memori

Untuk mendapatkan unjuk kerja yang baik, perlu diperhatikan benar-benar masalah manipulasi memori ini. Karena pada pemrograman grafik membutuhkan memori yang cukup besar bila dibandingkan dengan pemrograman dalam mode teks.

Untuk itu dalam program yang dibuat akan banyak digunakan teknik dinamik memori, yaitu menggunakan array dinamik. Array dinamik ini cukup fleksibel untuk dipakai pada pemrograman grafik, karena mudah untuk dibebaskan dan dialokasikan kembali.

4.3. Perangkat Keras dan Mode Tampilan

Program ini dibuat untuk dapat dijalankan pada komputer IBM PC/XT standart atau keluarganya.

Jenis tampilan yang dibutuhkan adalah tampilan dengan adapter CGA (Color Graphics Adapter). Sedangkan mode tampilan yang akan dipakai adalah mode graphics dengan resolusi 320 x 200 yang bisa memberikan fasilitas warna sebanyak 4 warna dalam satu palette.

Program grafik yang dibuat ini ditunjang oleh dua peralatan input, yaitu :

- a. File dari diskette
- b. Perangkat keras yang dibuat (lihat Bab 3)

File merupakan fasilitas input data yang sudah tersedia di dalam diskette. Sedangkan perangkat keras yang dibuat beserta papan grafik yang tersedia merupakan fasilitas input untuk pengeditan yang langsung dilakukan oleh pemakai.

4.4. Perangkat Lunak

Sebelum program ini dibuat maka harus direncanakan dulu bahasa pemrograman yang akan dipakai. Dari sekian banyak bahasa pemrograman yang ada ternyata yang memenuhi kriteria-kriteria diatas adalah bahasa C. Kriteria yang paling penting adalah karena pada bahasa C disediakan fasilitas manipulasi memori yang sangat memadai.

Kemudian dipilih compiler yang akan digunakan. Turbo C Versi 2.0 adalah compiler C yang sangat populer karena cara pengoperasiannya yang mudah. Selain itu Turbo C Versi 2.0 juga menyediakan fasilitas grafik.

4.5. Fasilitas Bagi Pemakai

Program ini dirancang untuk memberikan kemudahan kepada pemakai dalam membuat suatu disain grafik.

Fasilitas grafik yang diberikan oleh program grafik ini adalah:

- Mem-plot suatu titik dengan warna latar depan
- Menghapus titik dengan warna latar belakang
- Menggambar kotak dengan menggunakan 2 buah titik

- Menggambar garis dengan menggunakan 2 buah titik
- Menggambar lingkaran atau ellips dengan menggunakan 3 titik
- Menghapus gambar yang sudah diblok sebelumnya
- Mencat gambar dengan warna latar belakang
- Memblok gambar dan atau tanpa meng-copy-kannya ke tempat lain
- Mem-plot suatu daerah dengan warna tertentu.

4.6. Format File Gambar

Penyimpanan gambar dilakukan berdasarkan "bit mapped", jadi format file ini mudah dikonversikan ke dalam format file gambar untuk program grafik yang lain.

Data yang disimpan pada file gambar ini berbentuk data biner.

Untuk membedakannya dengan file-file lain maka pada awal file diberikan kunci file sejumlah 7 byte data, yang berupa karakter "Armygbr".

4.7. Penjelasan Program

4.7.1. Program Utama

Program utama merupakan bagian yang paling penting, karena kontrol program akan diatur melalui program utama ini. Setelah melakukan pendefinisian variabel, merubah sistem tampilan ke sistem grafik serta me-load gambar-gambar menu dari file maka untuk selanjutnya program akan menunggu salah satu tombol

keyboard ditekan. Tombol yang boleh ditekan hanya ada 4 pilihan, yaitu : tombol F2, SPACE, RETURN dan ESCAPE.

Berikut ini akan dijelaskan fungsi dari masing-masing tombol tersebut serta proses yang akan dijalankan.

- Jika tombol F2 yang ditekan :

Tombol F2 berfungsi untuk menampilkan menu utama bila mula-mula berada pada mode full screen, serta mengijinkan pemilihan menu utama. Akan tetapi menu utama tidak bisa ditampilkan pada semua kondisi. Ada satu kondisi di mana menu utama tidak bisa ditampilkan, yaitu bila saat itu blok sedang aktif dan salah satu sisi dari blok tersebut terletak antara baris 0 sampai 32. Sebelum menampilkan menu utama, maka tampilan yang ada pada layar akan disimpan pada buffer dinamik. Karena berupa buffer dinamik, maka terlebih dahulu buffer tersebut harus dialokasikan dulu sebesar data yang akan disimpan. Baru kemudian data gambar dimasukkan ke dalam buffer. Bila mula-mula layar dalam keadaan layar penuh (full-screen) maka akan dilakukan dua kali penyimpanan data. Pertama, akan menyimpan data gambar bagian atas yaitu yang berada pada baris 0 - 31, kemudian menyimpan data yang di bagian bawah yaitu yang berada pada baris 32 - 319. Namun, jika mula-mula menu utama sudah terampilkan maka hanya perlu dilakukan penyimpanan data gambar yang ada di bawah gambar menu utama.

Selanjutnya program utama akan memanggil fungsi PilihMenu. Setelah melakukan fungsi ini, akan didapatkan nilai variabel pilihan yang baru. Selain itu ada kemungkinan adanya perubahan status layar (full-screen atau menu utama aktif). Status ini diperlukan untuk mengembalikan tampilan. Yang tidak kalah pentingnya adalah mengembalikan kursor pada posisi semula.

- Jika tombol RETURN yang ditekan :

Tombol RETURN ini berfungsi untuk mengaktifkan menu yang dipilih apabila mula-mula menu tersebut tidak aktif. Dan juga untuk mengaktifkan posisi kursor saat itu, bila menu pada saat itu adalah salah satu dari menu no 4, 5, 6, 8 atau 9. Pengaktifan titik pada posisi kursor dilakukan dengan memanggil fungsi DefPoint.

- Jika tombol ESCAPE yang ditekan :

Tombol ESCAPE ini berfungsi untuk meng-non-aktif-kan menu yang terpilih apabila pada saat itu menu tersebut sedang aktif. Juga untuk membatalkan titik ke-1 atau titik ke-2 (yang diperlukan untuk menu 4, 5, 6, 8, 9) yang telah dipilih sebelumnya.

Jika pada saat itu status blok gambar sedang aktif maka blok akan dibatalkan.

- Jika tombol SPACE yang ditekan :

Tombol SPACE berfungsi untuk menterjemahkan dan menampilkan titik yang ada pada papan grafik sesuai

dengan posisi lengan bantu yang ada pada papan grafik ke dalam layar monitor. Pada papan grafik terdapat 2 potensiometer, sehingga ada 2 sinyal analog yang harus dikonversikan. Untuk itu pengkonversian dilakukan secara bergantian. Mula-mula input ke-1 dikonversikan terlebih dahulu, kemudian input ke-2. Untuk memilih input yang akan dikonversi, bisa dilakukan dengan membuat kombinasi logika bit ke 0,1 dan 2 dari port output 797. Bit ke-0 dihubungkan dengan Start of Conversion, bit ke-1 dihubungkan pada address A dan bit ke-2 dihubungkan dengan Output Enable. Langkah-langkah yang harus dilakukan untuk konversi adalah sebagai berikut :

- * Memberi sinyal positif edge pada pin Start of Conversion dari ADC dan sekaligus memilih address input analog yang akan dikonversikan. Untuk itu bisa digunakan perintah menulis data 8 bit pada port 697 (Heksa 2B9), atau `outportb(0x2B9,data)`.

- * Kemudian memonitor sinyal End of Conversion yang dihasilkan oleh ADC. Selama konversi, sinyal EOC akan berlogika 0. Namun sesaat setelah konversi selesai sinyal EOC akan berlogika 1. Untuk itu dilakukan perintah membaca data 8 bit pada port 697. Perintah ini akan terus dilakukan sampai data yang didapat dari port 697 berlogika 1. Berarti akan terjadi proses pengulangan perintah `inport(0xB9)`.

Karena pada perangkat keras yang dibuat digunakan sinyal clock yang tinggi untuk ADC nya, maka proses

pengkonversiannya menjadi sangat cepat. Sehingga tidak diperlukan lagi perintah untuk memonitor sinyal EOC.

* Setelah langkah di atas sudah dijalankan berarti data hasil konversi sudah ada dalam latch ADC 0809. Langkah selanjutnya yang perlu dikerjakan adalah mengirimkan sinyal berlogika 1 pada pin OE. Langkah ini dilakukan dengan memberikan perintah pengiriman sinyal berlogika '1' pada port output 697 output(0x2B9).

* Dan selanjutnya tinggal membaca data 8 bit hasil konversi dari port 696 (Hexa 2B8).

Data 8 bit hasil konversi tersebut belum sepenuhnya mencerminkan posisi dari lengan bantu. Dengan memasukkannya ke dalam suatu persamaan matematis, akan didapat sudut yang dibentuk oleh lengan bantu. Untuk lebih jelasnya lihat gambar 4.2.

Untuk mendapatkan koordinat X, Y dari lengan bantu, 2 buah data 8 bit hasil konversi (V1 & V2) harus diubah dengan persamaan berikut :

$$\theta_1 = \text{skala1} * V1 + \text{trans1}$$

$$\theta_2 = \text{skala2} * V2 + \text{trans2}$$

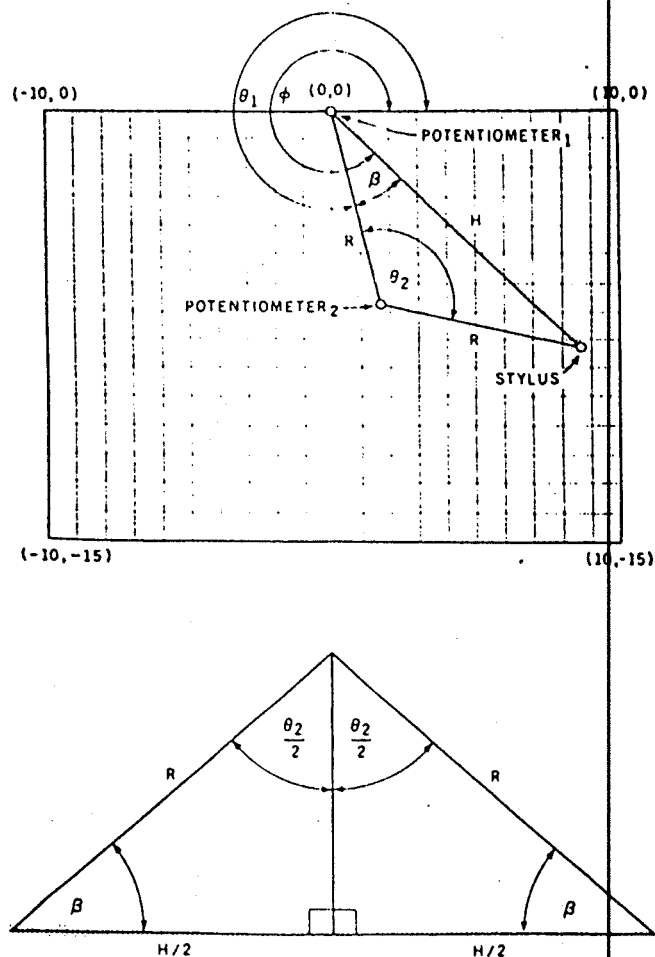
Dari bab 5 didapatkan :

$$\theta_1 = (0.8661421 * V1 - 118.07947) \text{ derajat}$$

$$\theta_2 = (- 0.85865 * V2 + 158.862429) \text{ derajat}$$

Dari gambar 4.2. diperoleh persamaan :

$$\beta = 90 - \theta_2/2 \text{ dan}$$



GAMBAR 4.2.

KONSEP TRIGONOMETRINYA

$$H/2R = \sin (\theta_2/2)$$

$$H = 2R \sin (\theta_2/2) , \text{ jadi}$$

$$\theta_2 = 2 \arcsin (h/2R)$$

Sedangkan sudut ϕ :

$$\phi = \theta_1 + \beta$$

Dengan persamaan untuk β :

$$\phi = \theta_1 + 90 - \theta_2/2$$

Jadi, X dan Y bisa dicari :

$$X = H \cos (\phi)$$

$$Y = H \sin (\phi)$$

Langkah-langkah perhitungannya, sebagai berikut :

- * Konversikan data analog sehingga didapatkan X dan Y
- * Dengan konstanta yang didapat dari Bab 5 bisa dicari θ_1 dan θ_2
- * R merupakan panjang lengan bantu pada papan grafik
- * Sehingga H dan ϕ dapat dihitung
- * Dengan demikian X dan Y bisa dicari.

Setelah X dan Y didapatkan, maka akan dicetak pada window yang ada pada menu utama bila menu utama aktif. Bila pada saat itu pilihan dari menu utama yang dipilih adalah no 2, 3 dan 7 serta pilihan tersebut sedang aktif, maka pilihan tersebut akan dijalankan pada titik X,Y. Selanjutnya pada titik X,Y akan dicetak kursornya. Khusus untuk pilihan = 3 (yaitu menu Curve Fitting) dibutuhkan lebih dari 3 titik. Setiap titik yang telah dipilih akan ditandai. Untuk membentuk kurva dari titik-titik yang telah dipilih tadi bisa dilakukan dengan menekan tombol End. Kurva yang terjadi adalah parabola derajat 5.

4.7.2. Fungsi SetKeGrafik

SetKeGrafik berfungsi untuk menset mode tampilan ke mode grafik. Fungsi ini intinya terletak pada instruksi :

```
initgraph( &GraphDriver, GraphMode, "" );
```

Instruksi initgraph akan menginisialisasi sistem grafik dengan me-load suatu driver grafik yang telah disediakan

(yang telah ada) pada disk dengan jalan mengalokasikan memori untuk driver tersebut, dan kemudian menempatkan sistem ke dalam mode grafik.

Instruksi `initgraph` harus dipanggil pertama kali untuk masuk ke dalam mode grafik. Parameter yang harus diikuti sertakan dalam instruksi ini adalah :

`GraphDriver`, `GraphMode` dan `PathDriver`

`GraphDriver` adalah parameter driver yang didukung oleh perangkat keras display.

Sedangkan `GraphMode` adalah mode untuk driver yang sudah dipilih.

Di dalam program yang dibuat untuk Tugas Akhir ini parameter-parameter yang dipakai adalah :

`GraphDriver = 1`

`GraphMode = 1`

Dengan driver CGA maka akan diperoleh 4 jenis palette.

Instruksi `initgraph` akan menghasilkan suatu nilai. Jika `initgraph` sukses maka nilai yang dihasilkan adalah 0. Namun jika terjadi kesalahan maka akan menghasilkan salah satu dari angka berikut -2, -3, -4, -5. Nilai kesalahan ini bisa diperoleh dengan instruksi:

`ErrorCode = graphresult()`

Untuk program ini, bila terjadi kesalahan pada saat inisialisasi ke mode grafik, maka program tidak akan melanjutkan proses (keluar ke DOS).

Bila inisialisasi sukses maka selanjutnya adalah men-set jenis teks yang akan dipakai dengan instruksi :

```
settextstyle(SMALL_FONT, HORIZ_DIR, 4)
```

Dari instruksi di atas terlihat:

- a. SMALL_FONT adalah jenis huruf yang dipakai.

Jenis huruf SMALL_FONT ini ditunjang oleh file LITT.CHR yang harus ada pada diskette.

- b. HORIZ_DIR adalah arah penulisan teks

Arah penulisan teks ini dipilih horizontal (dari kiri ke kanan).

- c. 4 adalah ukuran huruf SMALL_FONT yang akan digunakan

4.7.3. Fungsi Kursor

Kursor berfungsi untuk menggambar kursor pada layar. Sebelum menggambar kursor, terlebih dahulu harus diset segala sesuatunya, yaitu: jenis garis, jenis fill, mode penggambaran dan warna.

Bentuk kursor dibedakan dalam beberapa jenis, yaitu:

- a. Kursor untuk pilihan menu yang tidak diaktifkan
- b. Kursor untuk pilihan menu yang aktif

Untuk pilihan menu yang aktifpun, juga dibedakan berdasarkan pilihan menunya.

Pada bagian akhir dari fungsi ini adalah mengembalikan segala sesuatu yang telah diset pada awal fungsi pada keadaan semula.

4.7.4. Fungsi Cetak_Koord

Cetak_koord berfungsi untuk mencetak harga x dan

harga y pada window yang ada di menu utama.

Fungsi ini hanya bisa dilakukan bila gambar menu utama sedang ditampilkan (tidak dalam kondisi full screen). Untuk mencetak koordinat, warna yang diperlukan adalah hitam.

Pada saat gambar menu sedang aktif maka viewport sebelumnya pasti diset untuk daerah (0,32) sampai (319,199). Jadi, tanpa mengubah viewport tidak mungkin dapat menuliskan koordinat pada window yang ada di menu utama, karena window tersebut berada di luar daerah viewport.

Untuk itu viewport harus diset pada daerah (0,0) sampai (319,199) dengan instruksi:

```
setviewport(0, 0, 319, 199)
```

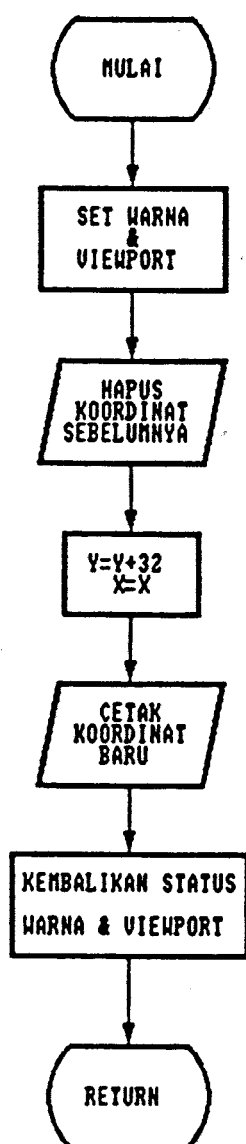
Teknik penulisan koordinat adalah dengan menghapus koordinat sebelumnya dan kemudian menuliskan koordinat yang baru. Perlu diketahui bahwa harga ordinat (y) harus ditambah dengan 32.

Penulisan koordinat hanya bisa dilakukan pada saat gambar menu aktif. Pada saat itu viewport diset pada daerah (0,32) sampai (319,199).

Untuk viewport mulai (0,32) sampai (319,199) maka posisi titik (0,0) dalam sistem grafik yang ada pada Turbo C V. 2.0 terletak pada posisi (0,32) di layar yang sebenarnya. Oleh sebab itu untuk memperoleh koordinat titik yang sebenarnya, maka y harus ditambah dengan 32.

4.7.5. Fungsi Load_menu

Load_menu berfungsi untuk mengalokasikan seluruh array dinamik yang akan dipakai, serta untuk memasukkan data gambar menu utama ke dalam array dan menampilkannya ke layar.



GAMBAR 4.3.

FLOWCHART FUNGSI CETAK_KOORD()

Pada bagian awal dari fungsi ini dilakukan pembukaan seluruh file yang berisi data gambar menu.

File `mnmain.gbr` berisi data gambar untuk menu utama, file `mnpic.gbr` berisi data gambar untuk menu file dan file `mntool.gbr` berisi data gambar untuk menu tool.

Proses selanjutnya adalah alokasi array dinamik. Array dinamik yang dialokasikan adalah :

- `buf_gbr1` untuk menyimpan data gambar pada daerah (0,0) sampai (319,31);
- `buf_gbr2` untuk menyimpan data gambar pada daerah (0,32) sampai (319,199);
- `bentuk1` untuk menyimpan data penunjuk menu utama;
- `bentuk2` untuk menyimpan data penunjuk menu file;
- `bentuk3` untuk menyimpan data penunjuk menu tool;
- `bentuk4` untuk menyimpan data penghapus window pilih warna dan pilih fill;
- `bentuk5` untuk menyimpan data penghapus window koordinat kursor;
- `tanda1` untuk menyimpan data penunjuk status pada menu tool;
- `tanda2` untuk menyimpan data penghapus penunjuk status pada menu tool;
- `buf_menu` untuk menyimpan data gambar menu utama.

4.7.6. Fungsi DefPoint

`DefPoint` berfungsi untuk menentukan status dari titik yang dipilih. `DefPoint` hanya dijalankan untuk pilihan gambar kotak, gambar garis, gambar lingkaran,

fill, blok gambar.

Titik yang bisa ditentukan statusnya adalah titik yang berada pada daerah $(0,0)$ sampai $(\text{max_x}, \text{max_y})$.

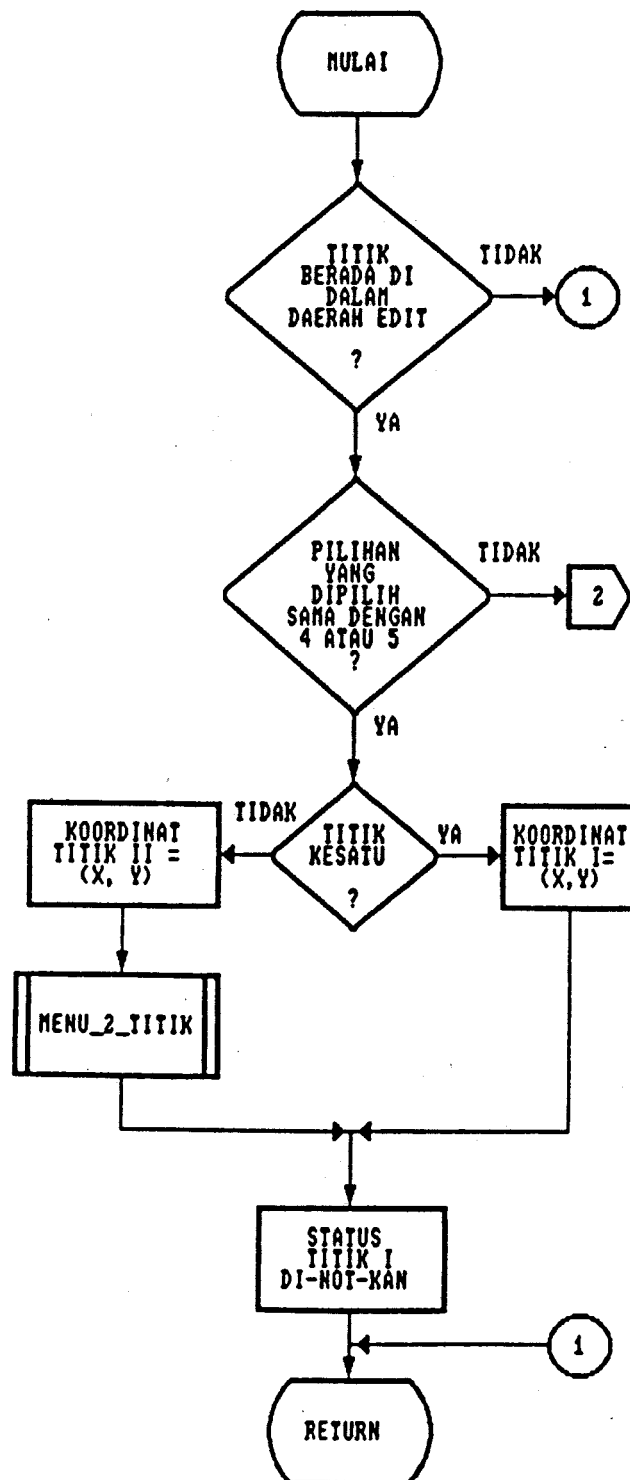
Bila pilihan = 4 atau 5:

- Untuk menggambar kotak dan gambar garis dibutuhkan dua buah titik (titik kesatu dan titik kedua).
- Dilakukan pengecekan status `first_point`. Status `first_point` ini menunjukkan apakah titik tersebut merupakan titik awal atau tidak.

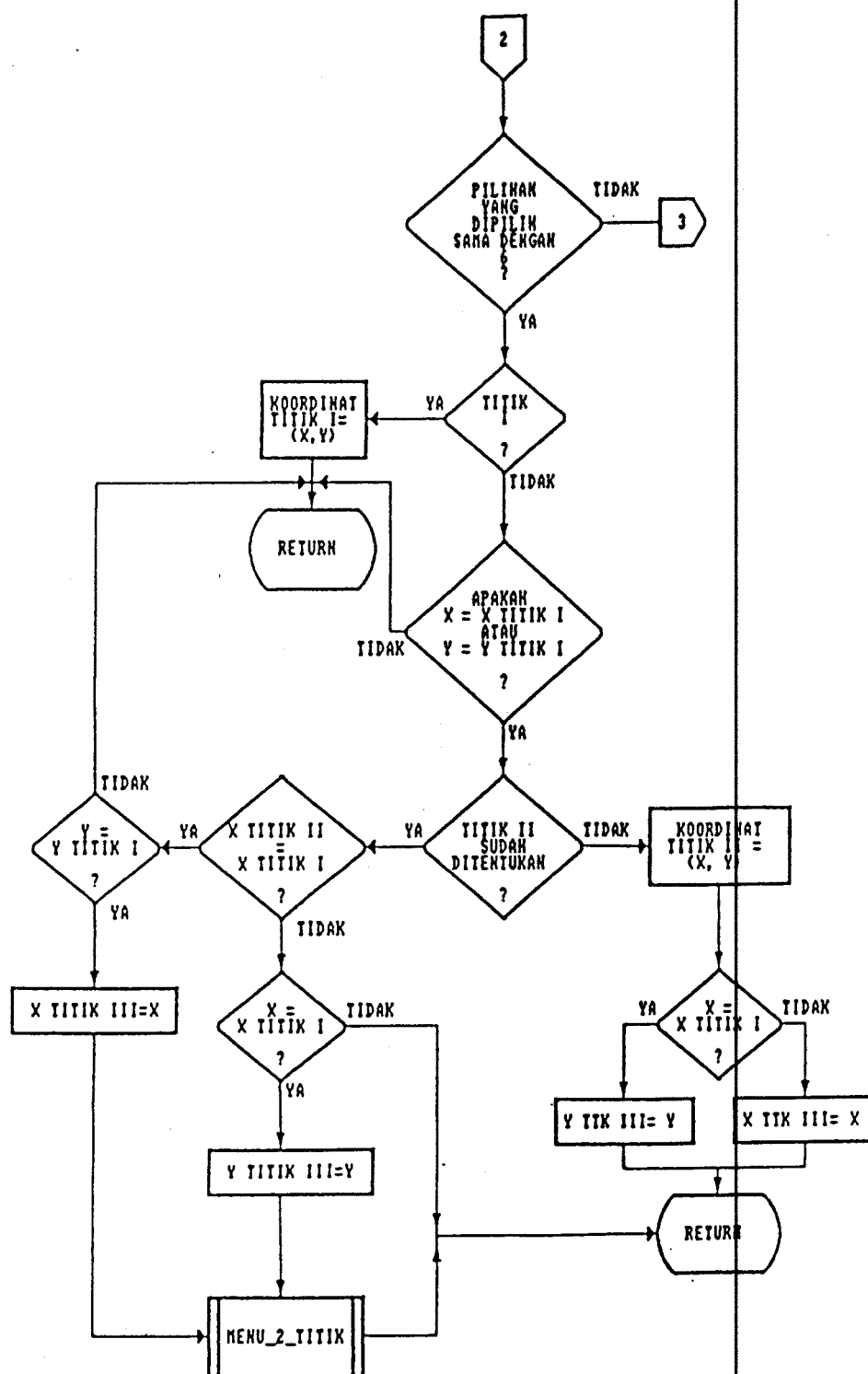
Bila `first_point` = 1 berarti titik tersebut adalah titik kesatu. Untuk itu, kemudian koordinat titik tersebut dijadikan koordinat titik awal (`start_x`, `start_y`). Bila `first_point` = 0 berarti titik tersebut merupakan titik kedua. Untuk itu, kemudian koordinat titik tersebut dijadikan koordinat titik akhir (`end_x`, `end_y`). Barulah setelah itu dipanggil fungsi `Menu_2_Titik`.

Bila pilihan = 6:

- Untuk menggambar lingkaran dibutuhkan 3 buah titik, dimana titik kesatu merupakan titik pusat lingkaran, sedangkan titik kedua dan titik ketiga merupakan titik yang ada di keliling lingkaran. Oleh sebab itu absis atau ordinat dari 2 buah titik yang terakhir harus sama dengan absis atau ordinat titik kesatu.
- Dilakukan pengecekan status `first_point`. Bila `first_point` = 1 berarti titik tersebut merupakan titik

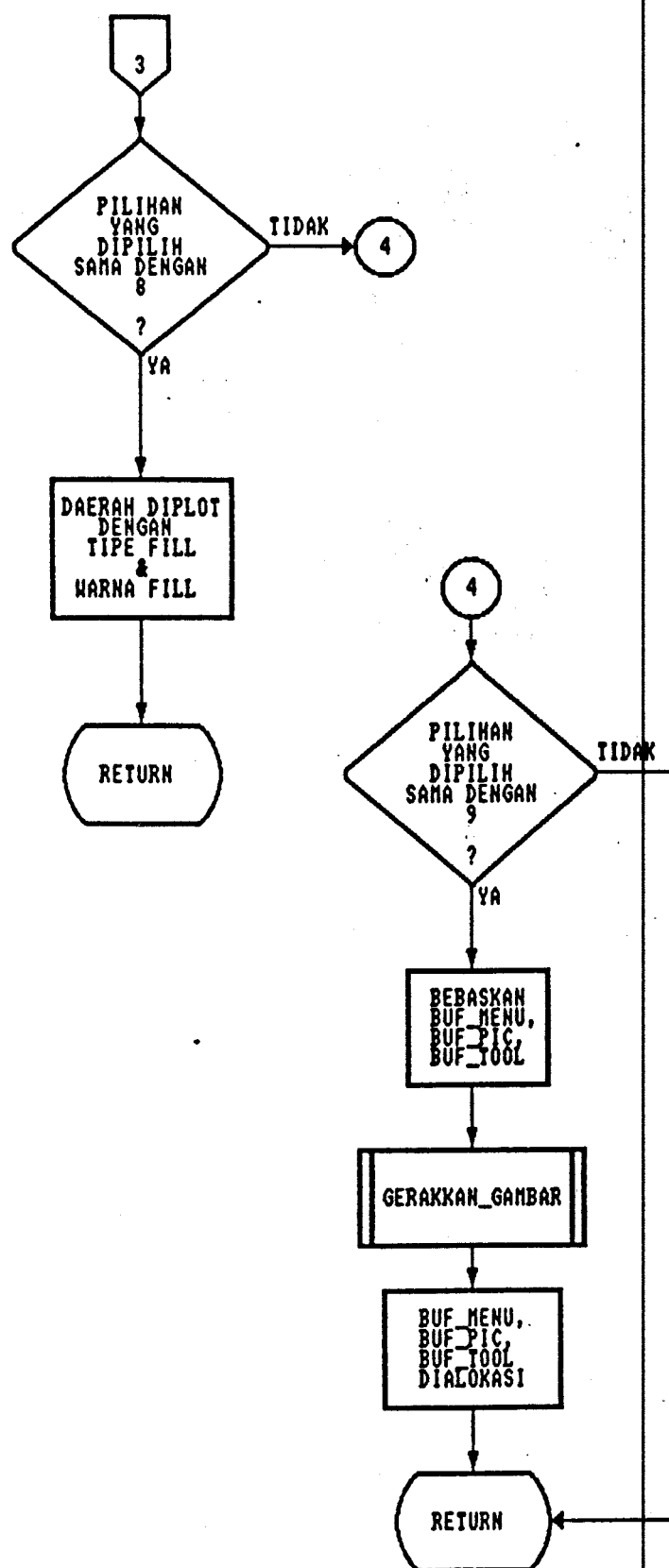


GAMBAR 4.4.
FLOWCHART FUNGSI DEFPOINT()



GAMBAR 4.5.

LANJUTAN KE-1 FLOWCHART FUNGSI DEFPOINT()



GAMBAR 4.6.

LANJUTAN KE-2 FLOWCHART FUNGSI DEFPPOINT()

awal. Untuk itu koordinat titik tersebut dijadikan koordinat titik awal (`start_x`, `start_y`). Bila `first_point = 0` berarti titik tersebut merupakan titik kedua, lalu dilakukan pengecekan lagi apakah salah satu komponen koordinatnya sama dengan komponen koordinat titik kesatu. Proses selanjutnya hanya bisa dilakukan bila salah satu komponen koordinatnya sama dengan komponen koordinat kesatu.

Setelah itu dilakukan pengecekan status `awal_nya`. Status `awal_nya` ini menunjukkan status titik kedua. Bila `awal_nya = 0` berarti titik kedua belum ditentukan. Bila `awalnya` tidak sama dengan 0 berarti titik kedua sudah ada. Untuk `awal_nya = 1` berarti harga `x` dari titik kedua sama dengan harga `x` dari titik kesatu, sehingga untuk selanjutnya dibutuhkan titik ketiga yang harga `y` nya sama dengan harga `y` dari titik kesatu. Untuk `awal_nya = 2` berarti harga `y` dari titik kedua sama dengan harga `y` dari titik kesatu, sehingga untuk selanjutnya dibutuhkan titik ketiga yang harga `x` nya sama dengan harga `x` dari titik kesatu.

Fungsi penggambaran yang sebenarnya yaitu `Menu_2_Titik` hanya bisa dilakukan bila syarat-syarat di atas sudah terpenuhi.

Bila pilihan = 8:

- Pilihan ini berfungsi untuk mem-fill suatu daerah. Untuk pilihan ini hanya diperlukan sebuah titik. Pilihan ini berfungsi untuk mem-plot suatu daerah yang dibatasi oleh warna gambar (`warna_gbr`) dengan

menggunakan warna fill (warna_fill) dan tipe fill (type_fill).

Bila pilihan = 9:

- Sebelumnya dilakukan pengecekan apakah blok sudah aktif atau belum. Bila blok belum diaktifkan (blok_aktif=0) maka kemudian dijalankan fungsi tentukan_titik_ke_2. Bila blok sudah diaktifkan maka fungsi yang dipanggil adalah gerakkan_gambar. Karena blok gambar membutuhkan array yang tidak sedikit maka sebelum fungsi gerakkan_gambar dijalankan harus dilakukan pembebasan array dinamik yang lain yaitu buf_menu, buf_pic dan buf_tool. Barulah setelah fungsi gerakkan_gambar selesai dijalankan, dilakukan lagi alokasi bagi array dinamik di atas.

4.7.7. Fungsi PilihMenu

PilihMenu berfungsi untuk memilih pilihan yang ada di menu utama maupun yang ada di menu file dan menu tool. Selain itu PilihMenu akan menandai pilihan yang dipilih.

Bila program ini baru saja dijalankan maka yang akan ditandai adalah pilihan ke 0. Nomor pilihan sebelumnya disimpan di variabel old_pilih. Old_pilih ini akan berguna bila ternyata pemilihan menu dibatalkan dengan menekan tombol escape.

Untuk memilih menu digunakan tombol panah kiri atau kanan. Sedangkan untuk dari menu ini bisa dilakukan

dengan menekan tombol return atau escape.

Bila return yang ditekan berarti menu yang dipilih akan diaktifkan, namun bila escape maka pilihan dibatalkan dan status menu kembali ke status semula. Untuk menunggu tombol ditekan, digunakan instruksi :

```
key.i = bioskey(0)
```

Nilai dari tombol yang ditekan disimpan pada variabel i yang ada pada struktur key. Tombol yang akan diantisipasi hanyalah tombol panah kiri dan kanan.

Bila tombol panah kiri ditekan maka pilihan dikurangkan satu dan bila tombol panah kanan, pilihan ditambah satu. Bila tombol kiri ditekan namun pilihan < 0 maka pilihan diberi harga maksimum yaitu 10, demikian juga bila tombol panah kanan ditekan tapi pilihan > 10 maka pilihan diberi harga minimum yaitu 0.

Untuk menandai pilihan yang dipilih digunakan teknik penghapusan tanda pilihan sebelumnya dan menandai pilihan pada saat itu.

Berikut ini penjelasan menu bila tombol return ditekan setelah melakukan pilihan.

a. Untuk pilihan = 1

Pilihan ini mempunyai menu tersendiri, yaitu menu file. Oleh sebab itu sebelumnya harus dilakukan penyimpanan gambar yang berada tepat pada lokasi menu file. Data gambar tersebut disimpan pada array dinamik `buf_sbl_menu`, yang tentu saja sebelumnya harus dialokasikan dulu. Array ini nanti akan dibebaskan setelah menu file selesai dijalankan.

Kemudian dilakukan pengecekan status awal_menu_pic. Status awal_menu_pic ini menunjukkan apakah sebelumnya menu file ini pernah diaktifkan atau tidak. Bila sebelumnya belum pernah diaktifkan maka dilakukan penyimpanan data gambar menu file dari file mnpic.gbr ke dalam array dinamik buf_pic. Baru kemudian gambar menu ditampilkan.

Pemilihan menu yang ada pada menu file ini dilakukan oleh PilihMenuLain. Namun sebelumnya status menu_lain_aktif dinolkan terlebih dahulu. Status ini digunakan untuk nencek apakah pilihan pada menu file tersebut aktif atau tidak.

Pilihan yang ada pada menu file adalah:

0. Load file gambar
1. Save file gambar
2. Print file gambar yang sudah di-load

Pada pilihan load dan save file gambar, segala sesuatunya dilaksanakan pada mode teks. Untuk itu status grafik harus diamankan dulu dengan instruksi simpan_status. Status ini nanti akan dikembalikan lagi dengan instruksi restore_status.

Instruksi restorecrtmode berfungsi untuk mengembalikan mode tampilan pada saat program ini akan dijalankan.

b. Untuk pilihan = 2

Pilihan ini berfungsi untuk mem-plot pixel pada posisi (x, y).

c. Untuk pilihan = 10

Proses yang terjadi pada bagian awal pilihan ini sama dengan pada pilihan = 1. Hanya saja, data gambar menu tool diambil dari file mntool.gbr dan dipindahkan ke array buf_tool.

Pada pilihan ini juga terdapat menu lain yaitu menu tool. Menu tool ini hanya bisa diaktifkan bila status menu_lain_aktif = 1.

Untuk pilihan menu tool = 1

Dipakai untuk mengganti status blok gambar menjadi terpotong. Artinya bila kita memblok gambar dan kemudian mengcopykannya ke tempat lain, maka blok gambar yang mula-mula akan dihapus. Status terpotong disimpan oleh variabel dipotong. Jadi, dipotong = 1.

Untuk pilihan menu tool = 2

Status blok gambar menjadi tidak terpotong. Jadi, dipotong = 0

Untuk pilihan menu tool = 3

Pilihan ini berfungsi untuk memilih warna latar belakang. Pemilihan warna bisa dilakukan dengan menekan tombol panah atas dan bawah. Warna yang dipilih akan ditampilkan pada window yang ada pada menu utama.

Untuk menunggu tombol ditekan digunakan instruksi: key2.i = bioskey(0)

Yang akan diantisipasi hanyalah tombol panah atas dan bawah. Bila tombol panah atas ditekan maka

warna_blk dikurangi satu, namun bila warna_blk ini kurang dari 0 maka warna_blk = 3.

Sebelumnya window pada menu utama dihapus dan kemudian diisi lagi dengan warna_blk yang dipilih.

Bila tombol panah bawah yang ditekan maka warna_blk ditambah satu, namun bila warna_blk ini lebih dari 3 maka warna_blk = 0.

Untuk pilihan menu tool = 4

Pilihan ini berfungsi untuk memilih warna latar depan (warna gambar).

Proses selanjutnya sama dengan pada pilihan menu tool = 3, hanya saja variabel yang bertambah dan berkurang adalah warna_gbr.

Untuk pilihan menu tool = 5

Pilihan ini berfungsi untuk memilih tipe dan warna fill. Pemilihan dilakukan dengan menekan tombol panah atas, bawah, kiri atau kanan.

Bila yang ditekan adalah tombol atas atau bawah maka pilih_fill dikurangi atau ditambah 1. Dengan harga minimum 0 dan harga maksimum 44.

Bila yang ditekan adalah tombol kiri atau kanan maka pilih_fill akan dikurangi atau ditambah 11.

Pada saat tombol kiri ditekan ternyata pilih_fill < 1 maka pilih_fill ditambah 44, dan bila saat tombol panah kanan ditekan ternyata pilih_fill > 44 maka pilih_fill dikurangi 44.

Untuk pilih menu tool = 6

Pilihan ini berfungsi untuk memilih warna dan tipe garis. Proses selanjutnya sama dengan pada pemilihan warna gambar atau warna latar belakang, hanya saja variabel yang berubah adalah pilih_garis dengan harga minimum 0 dan harga maksimum 7.

Selanjutnya dilakukan pengecekan pada variabel gbrmenu_aktif. Bila gambar menu tidak aktif maka data pada buf_gbr1 ditampilkan di layar di lokasi menu utama. Bila tidak maka pilihan pada menu utama akan ditandai dengan instruksi:

```
tampil_menu(0, pilihan)
```

Pada bagian akhir dari fungsi pilih_menu ini dilakukan pemindahan harga variabel kepada program pemanggilnya.

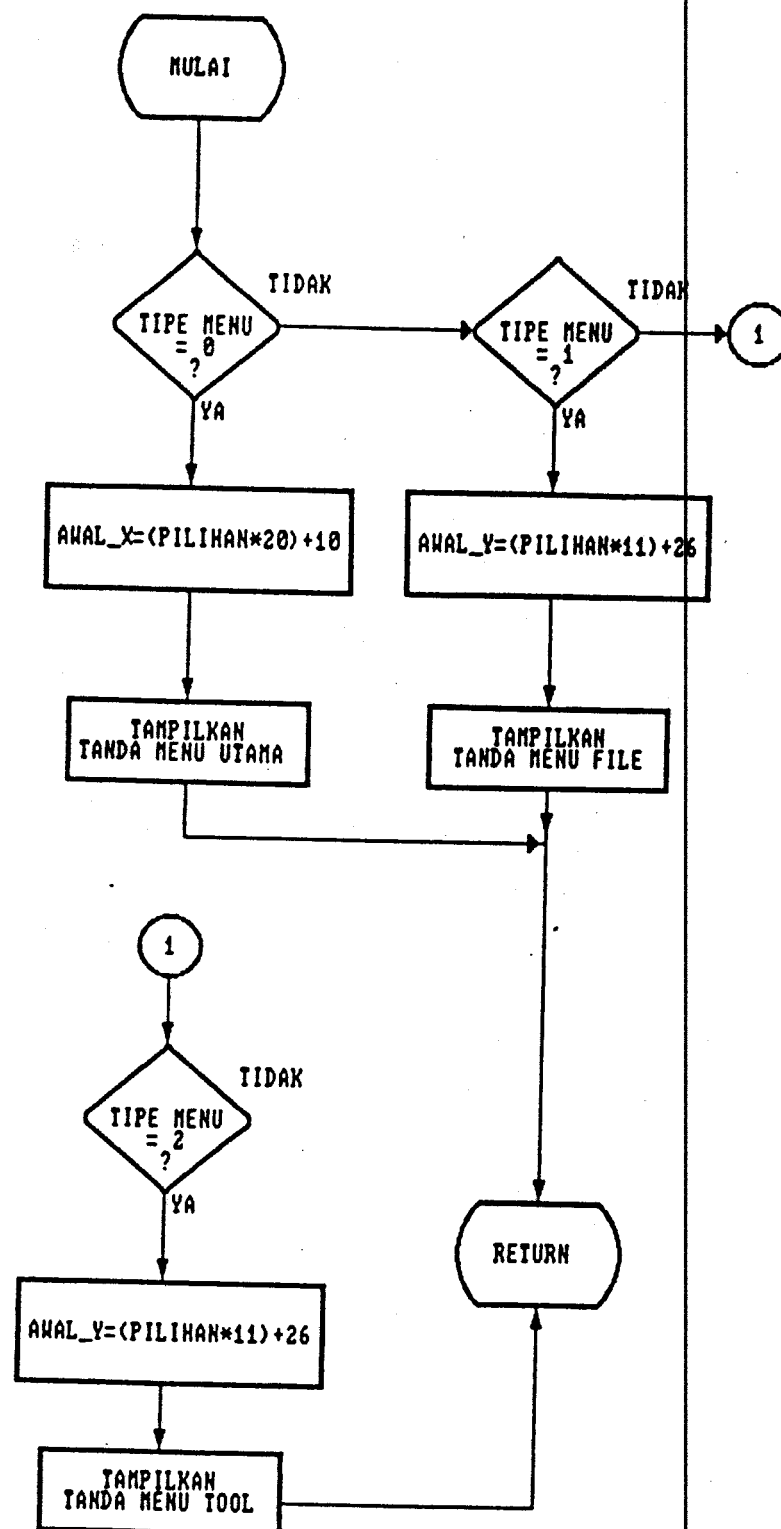
4.7.8. Fungsi Tampil_Menu

Tampil_Menu berfungsi untuk menandai pilihan, baik pada menu utama, menu file maupun menu tool.

Untuk memanggil fungsi ini diperlukan harga untuk tipe menu dan pilihan. Perincian tipe menu:

- Tipe menu 0 adalah menu utama
- Tipe menu 1 adalah menu file
- Tipe menu 2 adalah menu tool

Untuk menu utama, nomor pilihan berpengaruh pada absis dari posisi tanda. Posisi tanda menu terletak pada koordinat (awal_x, 6), dimana:



GAMBAR 4.7.

FLOWCHART FUNGSI TAMPIL_MENU()

```
awal_x = (pilihan * 20) + 10
```

Untuk menu file, posisi tanda menu terletak pada koordinat (28, awal_y), dimana:

```
awal_y = (pilihan * 11) + 26
```

Untuk menu tool, posisi tanda menu terletak pada koordinat (208, awal_y), dimana:

```
awal_y = (pilihan * 11) + 26
```

Tanda menu ditampilkan di layar dengan instruksi putimage.

4.7.9. Fungsi Menu_2_Titik

Menu_2_Titik berfungsi untuk menggambar kotak, garis dan lingkaran. Untuk memanggil fungsi ini diperlukan variabel koordinat awal, koordinat akhir, koordinat pada saat itu dan nomor pilihan.

Proses penggambaran kotak dilakukan dengan instruksi:

```
rectangle(x_mula, y_mula, x_akhir, y_akhir)
```

Proses penggambaran garis dilakukan dengan instruksi:

```
line(x_mula, y_mula, x_akhir, y_akhir)
```

Proses penggambaran lingkaran dilakukan dengan instruksi:

```
ellipse(x_mula, y_mula, 0, 360, delta_x, delta_y)
```

Dengan instruksi ini maka akan digambar ellips dengan titik pusat pada (x_mula, y_mula) mulai dari sudut 0 sampai sudut 360 (penuh) dengan radius x sebesar delta_x dan radius y sebesar delta_y.

4.7.10. Fungsi Restore

Restore berfungsi untuk memindahkan data gambar menu pada file kepada array dinamik yang telah ditentukan untuk kemudian ditampilkan di layar.

Untuk memanggil fungsi ini dibutuhkan variabel koordinat titik kiri atas dan kanan bawah gambar, nama file yang akan dibaca datanya serta nama array yang akan menyimpan data tersebut.

Pemindahan data gambar dari file dilakukan oleh instruksi: `*(buffer_nya + i) = getw(file_nya)`

4.7.11. Fungsi PilihMenuLain

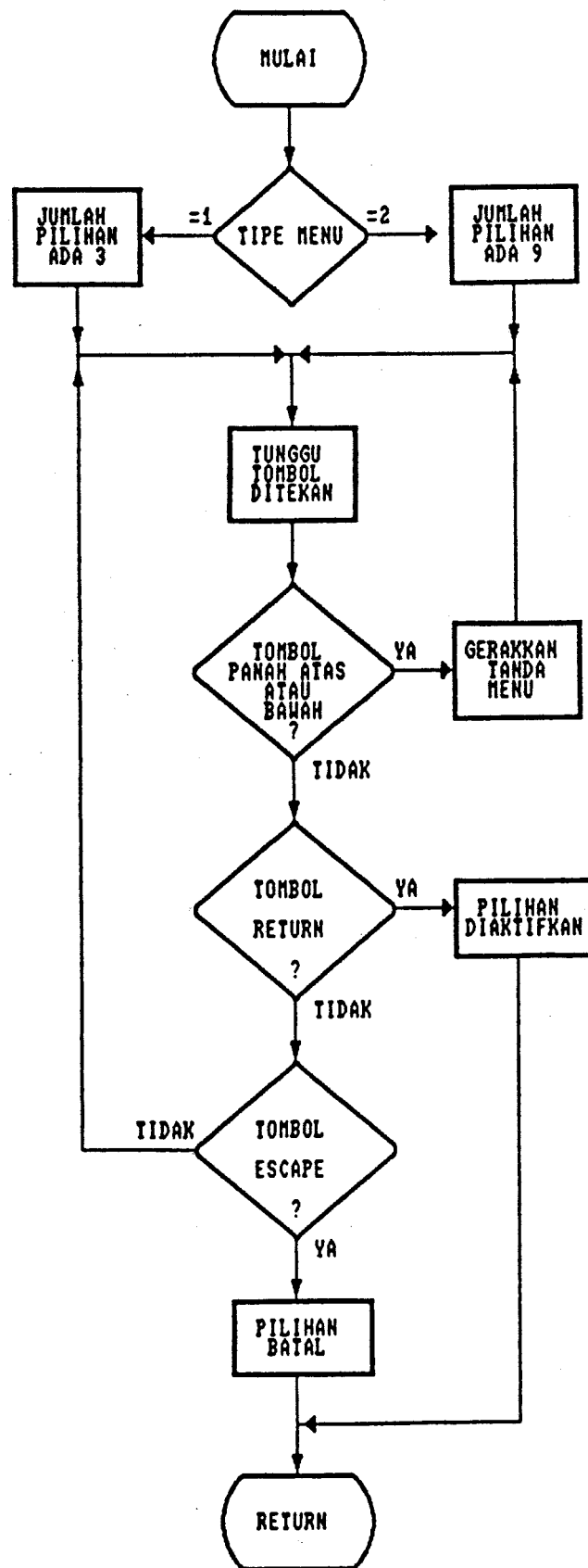
PilihMenuLain berfungsi untuk memilih pilihan yang ada pada menu file dan menu tool.

Untuk memanggil fungsi ini dibutuhkan variabel tipe menu. Pada menu file, pilihan yang disediakan sebanyak 3 buah dan pada menu tool pilihan yang disediakan sebanyak 7 buah. Pemilihan dilakukan dengan menekan tombol panah atas atau bawah.

Untuk menunggu penekanan tombol digunakan instruksi: `key.i = bioskey(0)`

Bila tombol yang ditekan adalah tombol panah atas maka pilihan_lain dikurangi satu namun bila pilihan_lain < 0 maka pilihan_lain = harga maksimumnya. Bila tombol yang ditekan adalah tombol panah bawah maka pilihan_lain ditambah satu.

Untuk memberi tanda menu dilakukan dengan



GAMBAR 4.8.

FLOWCHART FUNGSI PILIHMENULAIN()

menghapus tanda menu pada pilihan sebelumnya dan menampilkan tanda menu yang dipilih.

Pada bagian akhir dari fungsi ini dilakukan pengecekan, bila tombol yang ditekan adalah tombol return maka pilihan tadi diaktifkan. Namun bila escape yang ditekan maka pilihan tadi dibatalkan.

4.7.12. Fungsi Set_Garis

Set_Garis berfungsi untuk mengeset garis sesuai dengan pilihan yang dilakukan pada menu tool.

Untuk tipe garis no 4 tidak digunakan pada perangkat lunak grafik ini. Oleh sebab itu, jika pilihan_garis lebih dari 3 maka tipe garis dirubah menjadi 3.

Lebar garis yang disediakan oleh Turbo C V. 2.0 hanya ada 2 macam, yaitu lebar 1 pixel dan lebar 3 pixel.

4.7.13. Fungsi Set_Fill

Set_Fill berfungsi untuk mengeset fill sesuai dengan pilihan yang dilakukan pada menu tool.

Dari seluruh tipe fill yang disediakan oleh Turbo C V. 2.0., tipe no 12 tidak digunakan di program ini. Sedangkan warna fill sama seperti yang ada pada warna gambar, yaitu 4 buah warna.

Bila pilihan_fill antara 0 s/d 12 maka warna fill = 0

Bila pilihan_fill antara 11 s/d 23 maka warna fill = 1

Bila pilihan_fill antara 22 s/d 34 maka warna fill = 2

Bila pilihan_fill antara 33 s/d 45 maka warna fill = 3

Tipe fill ditentukan oleh persamaan di bawah ini:

$$\text{type_fill} = \text{pilihan_fill} - 11 * \text{warna_fill}$$

4.7.14. Fungsi Tentukan_Titik_Ke_2

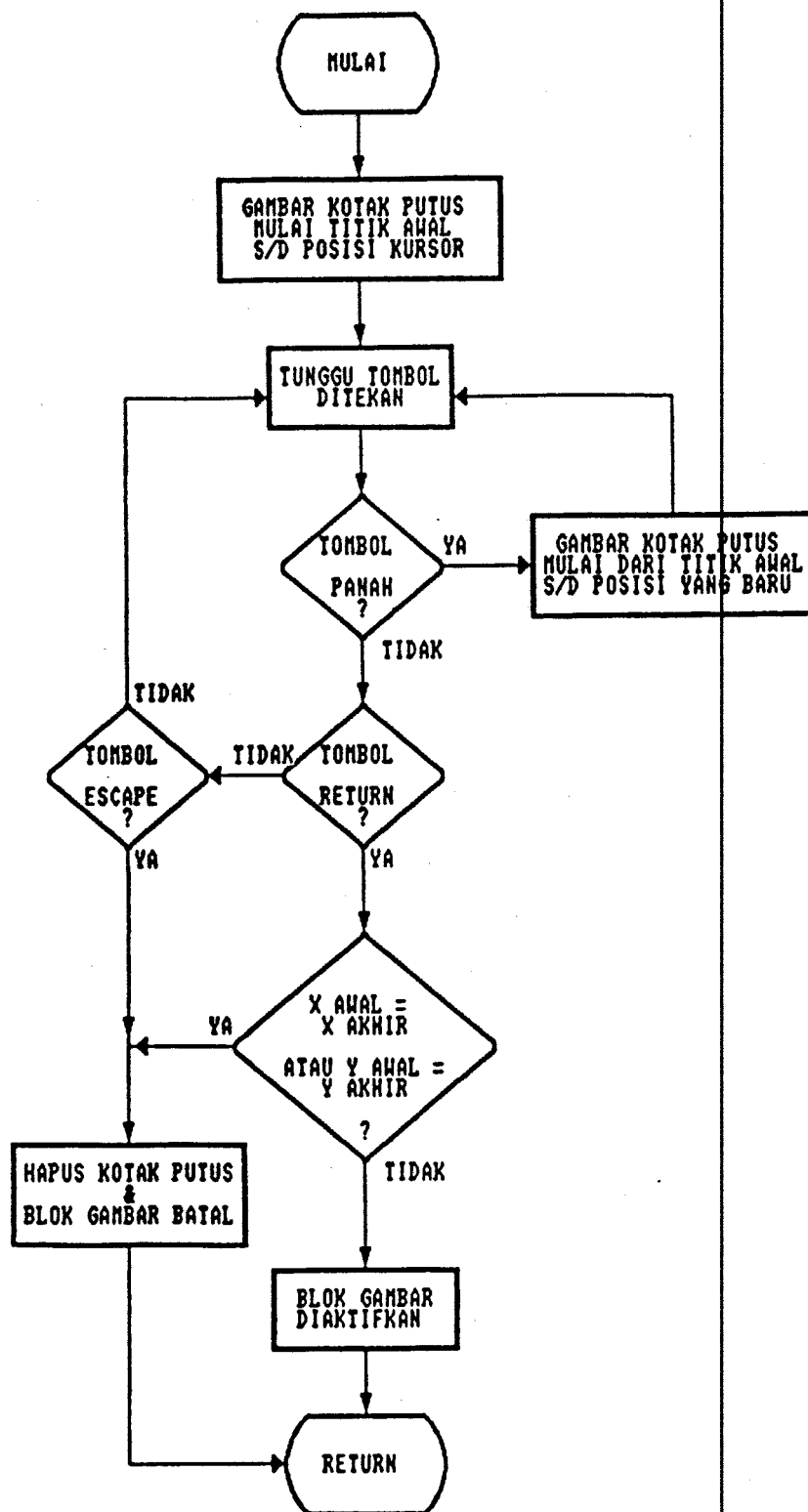
Tentukan_Titik_Ke_2 berfungsi untuk menggambar kotak putus-putus dan untuk menentukan titik kedua bagi blok gambar.

Untuk menggambar kotak putus-putus digunakan tombol panah atas, bawah, kiri dan kanan. Bila ditekan tombol panah atas atau bawah maka harga y dikurangi atau ditambah satu dengan daerah batas antara 0 s/d max_y . Bila ditekan tombol panah kiri atau kanan maka harga x dikurangi atau ditambah satu dengan daerah batas antara 0 s/d max_x .

Untuk keluar dari fungsi ini bisa dilakukan dengan tombol return atau escape. Bila escape yang ditekan maka blok gambar dibatalkan. Bila return yang ditekan maka blok gambar diaktifkan.

Pengaktifan blok hanya bisa dilakukan bila absis titik kedua tidak sama dengan absis titik kesatu dan ordinat titik kedua tidak sama dengan ordinat titik kesatu.

Penggambaran kotak putus-putus dilakukan dengan menghapus gambar kotak putus sebelumnya dan menggambar kotak putus dengan ukuran dan posisi yang baru. Untuk menggambar dan menghapus kotak putus digunakan fungsi kotak_putus.



GAMBAR 4.9.

FLOWCHART FUNGSI TENTUKAN_TITIK_KE_2()

4.7.15. Fungsi Kotak_Putus

Kotak_Putus berfungsi untuk menggambar kotak putus-putus. Instruksi yang dipakai:

```
rectangle(start_x, start_y, end_x, end_y)
```

4.7.16. Fungsi Cetak_Tanda

Cetak_Tanda berfungsi untuk mencetak tanda status pada menu tool. Yang dicetak adalah tanda untuk status full screen dan status blok gambar.

4.7.17. Fungsi Simpan_Status

Simpan_Status berfungsi untuk menyimpan status dari viewport dan mode grafik.

4.7.18. Fungsi Restore_Status

Restore_Status berfungsi untuk mengeset kembali status-status grafik seperti mode grafik, warna, tipe garis, tipe fill, tipe teks dan view port.

BAB V

PENUTUP

Untuk bisa menarik kesimpulan dari Tugas Akhir yang telah dibuat, maka tentunya terlebih dahulu harus dilakukan uji coba pada perangkat keras bersama-sama dengan perangkat lunaknya.

Uji coba ini dilakukan dalam rangka meng-kalibra-si perangkat keras yang dibuat.

Seperti dijelaskan pada Bab 4, bahwa hasil konversi sinyal analog menjadi data digital 8 bit pada dasarnya mencerminkan posisi dari lengan-lengan bantu yang ada pada papan grafik. Untuk itu data 8 bit tersebut harus diproses dulu agar didapatkan sudut yang dibentuk oleh lengan-lengan bantu terhadap sumbu koordinat (lihat gambar 4.2.). Data 8 bit itu harus dimasukkan dalam suatu persamaan yang mempunyai bentuk umum sebagai berikut :

$$\theta_1 = \text{skala}_1 * V_1 + \text{trans}_1$$

$$\theta_2 = \text{skala}_2 * V_2 + \text{trans}_2$$

dimana : θ_1 = sudut antara lengan 1 dan sumbu x

θ_2 = sudut antara lengan 1 dengan lengan 2

V_1 = data digital 8 bit hasil konversi input analog dari potensiometer ke-1

V_2 = data digital 8 bit hasil konversi input analog dari potensiometer ke-2

Skala dan Trans adalah konstanta-konstanta yang digunakan

Dalam pengukuran ini konstanta-konstanta skala dan trans tidak bisa langsung didapatkan.

Teknik pengukurannya seperti berikut ini :

- Lengan yang akan diukur digerakkan pada sudut tertentu (seperti pada tabel) sedangkan lengan yang tidak diukur ditempatkan pada posisi relatif yang tetap.
- Kemudian program pengukur dijalankan
- Setelah semua data V_1 dan V_2 untuk setiap sudut yang ditentukan telah didapatkan, maka dari data-data tersebut dicari persamaan regresi linier.
- Dari persamaan regresi linier bisa dicari konstanta skala dan trans.

Dengan menggunakan perangkat lunak Microstat didapatkan persamaan regresi linier sebagai berikut :

$$V_1 = 1.154545 * \theta_1 - 136.328063$$

$$\text{Sehingga : } \theta_1 = (0.8661421 * V_1 + 118.07947)^\circ$$

$$V_2 = -1.164642 * \theta_2 + 185.017857$$

$$\text{Sehingga : } \theta_2 = (-0.8586 * V_2 + 158.862429)^\circ$$

Berikut ini hasil pengukuran V_1 dan V_2 untuk θ_1 dan θ_2 :

TABEL 5.1.
HASIL PENGUKURAN

θ_1 (°)	V1	θ_2 (°)	V2
110	1	20	182
120	6	30	151
130	16	40	139
140	26	50	127
150	37	60	115
160	47	70	103
170	57	80	91
180	69	90	80
190	82	100	68
200	93	110	56
210	105	120	45
220	117	130	33
230	127	140	22
240	141	150	10
250	153	160	1
260	165		
270	177		
280	189		
290	200		
300	211		
310	222		
320	233		
330	244		
340	255		

6.1. Kesimpulan

Dari hasil pembahasan dan uji coba alat maka diperoleh beberapa kesimpulan, antara lain :

- Pada saat konversi sinyal analog menjadi data digital 8 bit masih saja terjadi kesalahan pengukuran, hal ini disebabkan karena adanya faktor kesalahan pada ADC 0809 yang digunakan (± 0.5 LSB).
- Dengan menggunakan sinyal clock yang semakin besar maka waktu konversi pun akan semakin singkat, sehingga tidak perlu lagi diadakan pengecekan sinyal EOC.
- Dengan membaca hasil pengukuran didapat bahwa hasil tersebut kurang linier, hal ini disebabkan karena kualitas potensiometer yang dipasangkan pada masing-masing lengan bantu kurang linier.
- Sedangkan untuk perangkat lunaknya, diperoleh kesimpulan bahwa untuk pemrograman grafik harus betul-betul diperhatikan masalah alokasi memori. Karena program grafik yang menggunakan metoda bit map seperti yang dipakai untuk menyelesaikan tugas akhir ini, membutuhkan memori yang cukup besar.

6.2. Saran

Dari kesimpulan yang telah diambil maka saran-saran yang diberikan untuk pengembangan lebih lanjut, adalah :

- Untuk mendapatkan hasil konversi yang lebih teliti,

perlu digunakan ADC dengan tingkat kesalahan yang lebih rendah lagi, namun tentu saja biasanya diimbangi dengan faktor harga yang lebih tinggi. Pada ADC yang dikenal adanya 3 jenis kesalahan, yaitu :

- a. Zero error
- b. Linearity error
- c. Full-scale error

Kualitas suatu ADC antara lain ditentukan oleh jumlah maksimum dari ketiga jenis kesalahan di atas (total unadjusted error). Untuk ADC yang digunakan dalam Tugas Akhir ini (ADC 0809), seperti disebutkan dalam buku data, harga total unadjusted error sebesar 1.25. Untuk pengembangan lebih lanjut, lebih baik menggunakan ADC yang mempunyai total unadjusted error lebih kecil.

- Agar didapatkan hasil konversi yang lebih linier, maka bisa digunakan potensiometer dengan kualitas yang lebih linier. Kekurang-linieran potensiometer yang dipakai pada Tugas Akhir ini bisa dilihat pada Tabel 5.1. Misalnya untuk potensiometer yang pertama. Apabila harga V_1 yang didapat dimasukkan ke dalam persamaan untuk θ_1 , maka θ_1 yang didapat tidak sama persis dengan harga θ_1 yang ada di tabel (untuk V_1 tersebut). Apabila harga yang di tabel semakin mendekati harga hasil persamaan, maka potensiometer tersebut semakin linier.
- Untuk perangkat lunak, agar bisa dilakukan penghematan pemakaian memori, sebaiknya digunakan metoda

penyimpanan data berdasarkan vektor. Untuk metoda penyimpanan data berdasarkan bit mapped, data grafik yang disimpan sebesar sekitar 16000 byte. Besarnya data grafik ini tidak memperhitungkan obyek yang telah dibuat pada layar. Apakah obyek yang dibuat berjumlah sedikit ataupun berjumlah banyak, data grafik yang akan tersimpan tetap sekitar 16000 byte. Jadi, seluruh isi memori video disimpan ke dalam file. Sedangkan untuk penyimpanan data berdasarkan vektor, akan terjadi penghematan lokasi baik di memori maupun di diskette. Karena pada metoda vektor ini, yang disimpan hanyalah parameter-parameter yang mengikuti suatu penggambaran. Misalnya, untuk penggambaran garis, maka yang disimpan adalah parameternya yang berupa titik awal garis, titik akhir garis dan warna garis.

DAFTAR PUSTAKA

1. Artwick, Bruce A., Microcomputer Displays, Graphics, and Animation, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1985.
2. Corporation, IBM, IBM PC/XT Technical Reference, IBM Corporation.
3. Corporation, INTEL, MCS-85 User's Manual, INTEL Corporation, Santa Clara, California, 1977.
4. Eggebrecht, Lewis C., Interfacing to the IBM Personal Computer, Howard W. Sams & Co., Indianapolis, 1987.
5. Hall, Douglas V., Microprocessors and Interfacing: Programming and Hardware, Mc. Graw Hill Book Co., Singapore, 1986.
6. Incorporated, Texas Instruments, Interface Circuits Data Book, Texas Instruments Incorporated, Dallas, Texas, 1987.
7. International, Borland, Turbo C Version 2.0 User's Guide, Borland International, Scott Valley, 1988.
8. International, Borland, Turbo C Version 2.0 Reference Guide, Borland International, Scott Valley, 1988.
9. Johnson, Nelson, Advance Graphics In C Programming and Techniques, Mc. Graw Hill Inc., Berkeley, California, 1987.
10. Jourdain, Robert, Programmer's Problem Solver for IBM PC, XT and AT, Brady Books, New York, 1986.
11. Kernighan, Brian W., The C Programming Language, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1978.

12. Newman, William M, Principles of Interactive Computer Graphics, Mc. Graw Hill Book C., Japan, 1981.
13. Norton, Peter, Programmer's Guide to The IBM PC, Washington, Microsoft Press, 1985.
14. Schildt, Herbert, C Power User's Guide, Mc.Graw Hill Inc., Berkeley, California, 1988.
15. Sheingold, Daniel H., Analog-Digital Conversion Notes, Analog Device Inc, 1977.

USULAN TUGAS AKHIR

1. JUDUL : INTERFACE PAPAN GRAFIK PADA KOMPUTER IBM PC.
2. RUANG LINGKUP : - Analisa Sistem
- Computer Graphic
- Pemrograman Komputer
- Organisasi Komputer
3. LATAR BELAKANG : Sejak dahulu sampai saat ini, gambar banyak digunakan orang. Terlebih sejak mulai ditemukannya komputer. Sejak saat itulah mulai diadakan penelitian-penelitian terhadap kemungkinan digunakannya komputer sebagai alat bantu untuk membuat gambar. Perkembangan berlanjut dengan pesat pada akhir-akhir ini di mana mulai beredarnya komputer mikro yang bisa digunakan

untuk berbagai bidang. Dari digunakannya komputer sebagai alat bantu membuat gambar atau lebih dikenal dengan CAD (Computer Aided Design) maka timbullah disiplin ilmu yang baru yaitu computer graphic. Untuk menggunakan komputer sebagai alat bantu untuk menggambar tentu dibutuhkan suatu perangkat lunak. Selain itu untuk memudahkan pemakaian perangkat lunak tersebut dibutuhkan suatu peralatan input grafik.

4. PENELAAHAN STUDI : Perangkat lunak grafik harus menyediakan banyak fasilitas untuk membuat suatu disain grafik, misalnya untuk menggambar garis, menggambar lingkaran, menggambar kotak dan lain lain. Tiap fungsi di atas mempunyai algoritma tersendiri. Algoritma ini dibutuhkan karena gambar (grafik) yang akan dibuat pada dasarnya akan diterapkan pada sistem digital. Sistem digital ini merupakan sistem yang diskrit (tidak kontinu). Oleh sebab itu dibutuhkan suatu algoritma untuk memilih pixel yang

akan di-on-kan atau di-off-kan sehingga seolah-olah membentuk suatu gambar yang tidak diskrit. Dengan adanya bahasa pemrograman yang juga memiliki fasilitas fungsi grafik, maka pembuatan perangkat lunak grafik akan menjadi lebih mudah.

5. TUJUAN : Membuat perangkat lunak grafik dengan teknik di atas, serta membuat interface yang bersama-sama dengan papan grafik yang tersedia akan bertindak sebagai peralatan input bagi perangkat lunak grafik di atas.

6. LANGKAH-LANGKAH :

1. Mempelajari literatur yang berhubungan dengan computer graphic dan interfacing pada komputer IBM PC.
2. Membuat perangkat lunak
3. Membuat perangkat keras
4. Uji coba perangkat keras bersama-sama dengan perangkat lunaknya.
5. Memperbaiki kesalahan pada perangkat lunak maupun keras.

6. Membuat kesimpulan dan menyusun buku.

7. JADWAL KEGIATAN :

KEGIATAN	BULAN KE					
	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

8. RELEVANSI : Diharapkan tugas akhir yang dibuat ini dapat dipakai untuk membantu pekerjaan pembuat disain atau siapa saja yang ingin menggunakannya.

ADC 0809

ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

D2042, JUNE 1981—REVISED FEBRUARY 1986

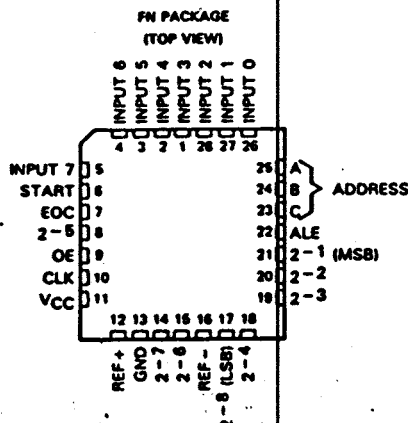
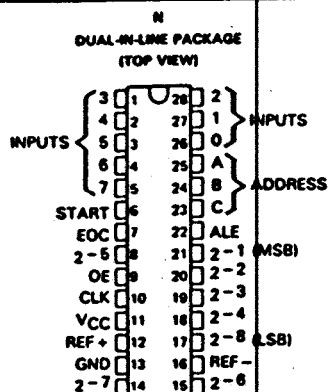
- Total Unadjusted Error . . . ± 0.75 LSB Max for ADC0808 and ± 1.25 LSB Max for ADC0809
- Resolution of 8 Bits
- 100 μ s Conversion Time
- Ratiometric Conversion
- Guaranteed Monotonicity
- No Missing Codes
- Easy Interface with Microprocessors
- Latched 3-State Outputs
- Latched Address Inputs
- Single 5-Volt Supply
- Low Power Consumption
- Designed to be Interchangeable with National Semiconductor ADC0808, ADC0809

description

The ADC0808 and ADC0809 are monolithic CMOS devices with an 8-channel multiplexer, an 8-bit analog-to-digital (A/D) converter, and microprocessor-compatible control logic. The 8-channel multiplexer can be controlled by a microprocessor through a 3-bit address decoder with address load to select any one of eight single-ended analog switches connected directly to the comparator. The 8-bit A/D converter uses the successive-approximation conversion technique featuring a high-impedance threshold detector, a switched-capacitor array, a sample-and-hold, and a successive-approximation register (SAR). Detailed information on interfacing to most popular microprocessors is readily available from the factory.

The comparison and converting methods used eliminate the possibility of missing codes, nonmonotonicity, and the need for zero or full-scale adjustment. Also featured are latched 3-state outputs from the SAR and latched inputs to the multiplexer address decoder. The single 5-volt supply and low power requirements make the ADC0808 and ADC0809 especially useful for a wide variety of applications. Ratiometric conversion is made possible by access to the reference voltage input terminals.

The ADC0808 and ADC0809 are characterized for operation from -40°C to 85°C .



2

Data Acquisition Circuits

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

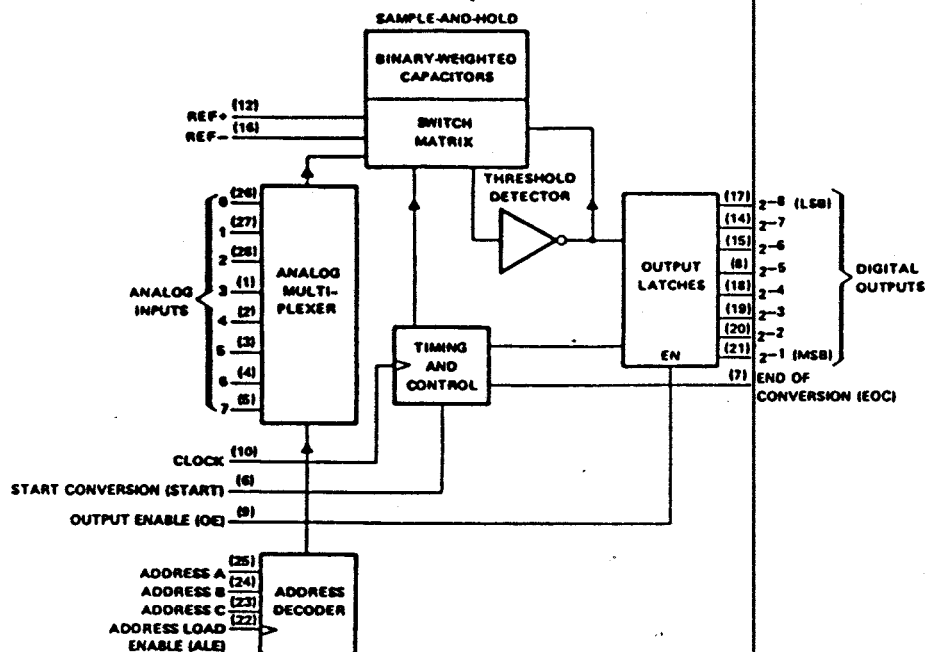
Copyright © 1983, Texas Instruments Incorporated

ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

functional block diagram (positive logic)

2

Data Acquisition Circuits



MULTIPLEXER FUNCTION TABLE

INPUTS				SELECTED ANALOG CHANNEL
ADDRESS		ADDRESS	STROBE	
C	B	A		
L	L	L	↑	0
L	L	H	↑	1
L	H	L	↑	2
L	H	H	↑	3
H	L	L	↑	4
H	L	H	↑	5
H	H	L	↑	6
H	H	H	↑	7

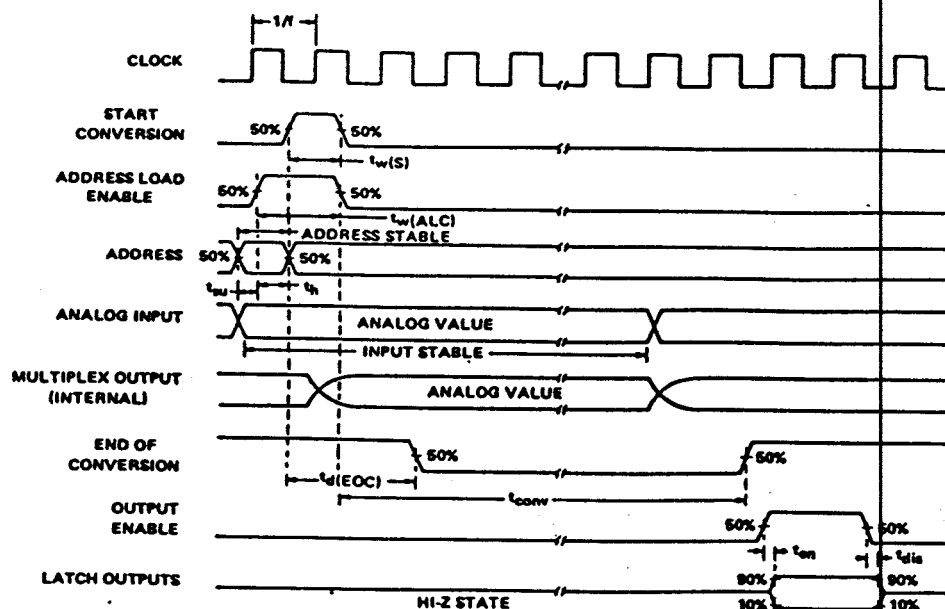
H = high level, L = low level
↑ = low-to-high transition

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

ADC0808, ADC0809
CMOS ANALOG-TO-DIGITAL CONVERTERS
WITH 8-CHANNEL MULTIPLEXERS

operating sequence



2

Data Acquisition Circuits

ADC0808, ADC0809

CMOS ANALOG-TO-DIGITAL CONVERTERS

WITH 8-CHANNEL MULTIPLEXERS

2

Data Acquisition Circuits

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	6.5 V
Input voltage range: control inputs	-0.3 to 15 V
all other inputs	-0.3 V to $V_{CC} + 0.3$ V
Operating free-air temperature range	-40°C to 85°C
Storage temperature range	-65°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to network ground terminal.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	6	V
Positive reference voltage, V_{ref+} (see Note 2)		V_{CC}	$V_{CC} + 0.1$	V
Negative reference voltage, V_{ref-}		0	-0.1	V
Differential reference voltage, $V_{ref+} - V_{ref-}$		5		V
High-level input voltage, V_{IH}	$V_{CC} - 1.5$			V
Low-level input voltage, V_{IL}			1.5	V
Start pulse duration, $t_w(S)$	200			ns
Address load control pulse duration, $t_w(ALC)$	200			ns
Address setup time, t_{su}	50			ns
Address hold time, t_h	50			ns
Clock frequency, f_{clock}	10	640	1280	kHz
Operating free-air temperature, T_A	-40		85	°C

NOTE 2: Care must be taken that this rating is observed even during power-up.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655312 • DALLAS, TEXAS 75265

ADC0808, ADC0809
CMOS ANALOG-TO-DIGITAL CONVERTERS
WITH 8-CHANNEL MULTIPLEXERS

electrical characteristics over recommended operating free-air temperature range, $V_{CC} = 4.75 \text{ V}$ to 5.25 V (unless otherwise noted)

total device

PARAMETER		TEST CONDITIONS	MIN	TYP [†]	MAX	UNIT
V_{OH}	High-level output voltage	$I_O = -360 \mu\text{A}$	$V_{CC} - 0.4$			V
V_{OL}	Low-level output voltage	Data outputs $I_O = 1.6 \text{ mA}$			0.45	V
		End of conversion $I_O = 1.2 \text{ mA}$			0.45	
I_{OZ}	Off-state (high-impedance-state) output current	$V_O = V_{CC}$			3	μA
		$V_O = 0$			-3	
I_I	Control input current at maximum input voltage	$V_I = 15 \text{ V}$			1	μA
I_{IL}	Low-level control input current	$V_I = 0$			-1	μA
I_{CC}	Supply current	$f_{\text{clock}} = 640 \text{ kHz}$		0.3	3	mA
C_i	Input capacitance, control inputs	$T_A = 25^\circ\text{C}$		10	15	pF
C_o	Output capacitance, data outputs	$T_A = 25^\circ\text{C}$		10	15	pF
Resistance from pin 12 to pin 16				1000		k Ω

analog multiplexer

PARAMETER		TEST CONDITIONS	MIN	TYP [†]	MAX	UNIT
I_{on}	Channel on-state current (see Note 3)	$V_I = 5 \text{ V}$, $f_{\text{clock}} = 640 \text{ kHz}$			2	μA
		$V_I = 0$, $f_{\text{clock}} = 640 \text{ kHz}$			-2	
I_{off}	Channel off-state current	$V_{CC} = 5 \text{ V}$, $V_I = 5 \text{ V}$		10	200	nA
		$T_A = 25^\circ\text{C}$, $V_I = 0$		-10	-200	
		$V_{CC} = 5 \text{ V}$, $V_I = 5 \text{ V}$			1	μA
		$V_I = 0$			-1	

[†]Typical values are at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$.

NOTE 3: Channel on-state current is primarily due to the bias current into or out of the threshold detector, and it varies directly with clock frequency.

2

Data Acquisition Circuits

ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

operating characteristics, $T_A = 25^\circ\text{C}$, $V_{CC} = V_{REF+} = 5\text{ V}$, $V_{REF-} = 0\text{ V}$, $f_{\text{clock}} = 640\text{ kHz}$
(unless otherwise noted)

2

Data Acquisition Circuits

PARAMETER		TEST CONDITIONS	ADC0808			ADC0809			UNIT
			MIN	TYP†	MAX	MIN	TYP†	MAX	
kSVS	Supply voltage sensitivity	VCC = Vref+ = 4.75 V to 5.25 V, TA = -40°C to 85°C, See Note 4	±0.05			±0.05			%/V
	Linearity error (see Note 5)		±0.25			±0.5			LSB
	Zero error (see Note 6)		±0.25			±0.25			LSB
	Total unadjusted error (See Note 7)	TA = 25°C	±0.25 ±0.5			±0.5			LSB
		TA = -40°C to 85°C	±0.75			±1.25			
		TA = 0°C to 70°C				±1			
ten	Output enable time	CL = 50 pF, RL = 10 kΩ	80 250			80 250			ns
tdis	Output disable time	CL = 10 pF, RL = 10 kΩ	105 250			105 250			ns
tconv	Conversion time	See Note 8	90	100	116	90	100	116	μs
td(EOC)	Delay time, end of conversion output	See Notes 8 and 9	0 14.5			0 14.5			μs

[†]Typical values for all except supply voltage sensitivity are at $V_{CC} = 5\text{ V}$, and all are at $T_A = 25^\circ\text{C}$.

- NOTES:
- Supply voltage sensitivity relates to the ability of an analog-to-digital converter to maintain accuracy as the supply voltage varies. The supply and V_{REF+} are varied together and the change in accuracy is measured with respect to full-scale.
 - Linearity error is the maximum deviation from a straight line through the end points of the A/D transfer characteristic.
 - Zero error is the difference between 00000000 and the converted output for zero input voltage; full-scale error is the difference between 11111111 and the converted output for full-scale input voltage.
 - Total unadjusted error is the maximum sum of linearity error, zero error, and full-scale error.
 - Refer to the operating sequence diagram.
 - For clock frequencies other than 640 kHz, $t_{\text{d(EOC)}}$ maximum is 8 clock periods plus 2 μs .

0

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

PRINCIPLES OF OPERATION

The ADC0808 and ADC0809 each consists of an analog signal multiplexer, an 8-bit successive-approximation converter, and related control and output circuitry.

multiplexer

The analog multiplexer selects 1 of 8 single-ended input channels as determined by the address decoder. Address load control loads the address code into the decoder on a low-to-high transition. The output latch is reset by the positive-going edge of the start pulse. Sampling also starts with the positive-going edge of the start pulse and lasts for 32 clock periods. The conversion process may be interrupted by a new start pulse before the end of 64 clock periods. The previous data will be lost if a new start of conversion occurs before the 64th clock pulse. Continuous conversion may be accomplished by connecting the End-of-Conversion output to the start input. If used in this mode an external pulse should be applied after power up to assure start up.

converter

The CMOS threshold detector in the successive-approximation conversion system determines each bit by examining the charge on a series of binary-weighted capacitors (Figure 1). In the first phase of the conversion process, the analog input is sampled by closing switch S_C and all S_T switches, and by simultaneously charging all the capacitors to the input voltage.

In the next phase of the conversion process, all S_T and S_C switches are opened and the threshold detector begins identifying bits by identifying the charge (voltage) on each capacitor relative to the reference voltage. In the switching sequence, all eight capacitors are examined separately until all 8 bits are identified, and then the charge-convert sequence is repeated. In the first step of the conversion phase, the threshold detector looks at the first capacitor (weight = 128). Node 128 of this capacitor is switched to the reference voltage, and the equivalent nodes of all the other capacitors on the ladder are switched to REF^- . If the voltage at the summing node is greater than the trip-point of the threshold detector (approximately one-half the V_{CC} voltage), a bit is placed in the output register, and the 128-weight capacitor is switched to REF^- . If the voltage at the summing node is less than the trip point of the threshold detector, this 128-weight capacitor remains connected to REF^+ through the remainder of the capacitor-sampling (bit-counting) process. The process is repeated for the 64-weight capacitor, the 32-weight capacitor, and so forth down the line, until all bits are counted.

With each step of the capacitor-sampling process, the initial charge is redistributed among the capacitors. The conversion process is successive approximation, but relies on charge redistribution rather than a successive-approximation register (and reference DAC) to count and weigh the bits from MSB to LSB.

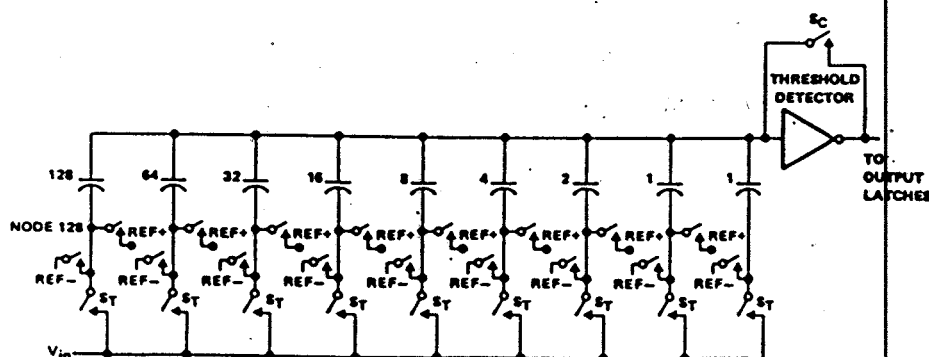


FIGURE 1. SIMPLIFIED MODEL OF THE SUCCESSIVE-APPROXIMATION SYSTEM

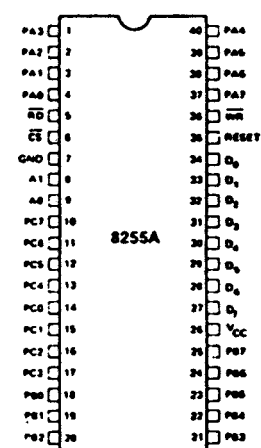
8255

8255A, 8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40 Pin Dual-In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in two groups of twelve and used in three major modes of operation. In the first mode (Mode 0), each group of twelve I/O pins may be programmed in sets of 4 to be input or output. In Mode 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining four pins three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a Bi-directional Bus mode which uses 8 lines for a bi-directional bus, and five lines, borrowing one from the other group, for handshaking.

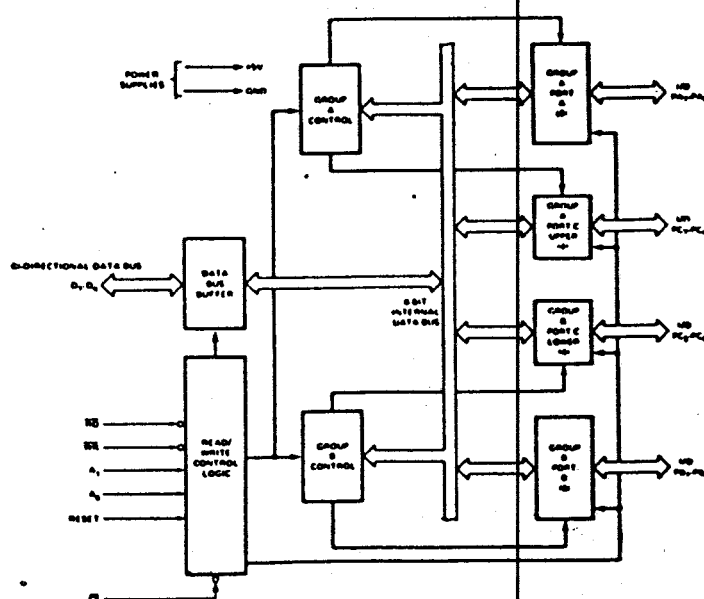
PIN CONFIGURATION



PIN NAMES

D ₀ -D ₇	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A ₀ , A ₁	PORT ADDRESS
PA ₇ -PA ₀	PORT A (8BIT)
PB ₇ -PB ₀	PORT B (8BIT)
PC ₇ -PC ₀	PORT C (8BIT)
V _{CC}	+5 VOLTS
GND	0 VOLTS

8255A BLOCK DIAGRAM



General

The 8255 is a Programmable Peripheral Interface (PPI) device designed for use in Intel Microcomputer Systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255 is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state, bi-directional, eight bit buffer is used to interface the 8255 to the system data bus. Data is transmitted or received by the buffer upon execution of INPUT or OUTPUT instructions by the CPU. Control Words and Status information are also transferred through the Data Bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(\overline{CS})

Chip Select: A "low" on this input pin enables the communication between the 8255 and the CPU.

(\overline{RD})

Read: A "low" on this input pin enables the 8255 to send the Data or Status information to the CPU on the Data Bus. In essence, it allows the CPU to "read from" the 8255.

(\overline{WR})

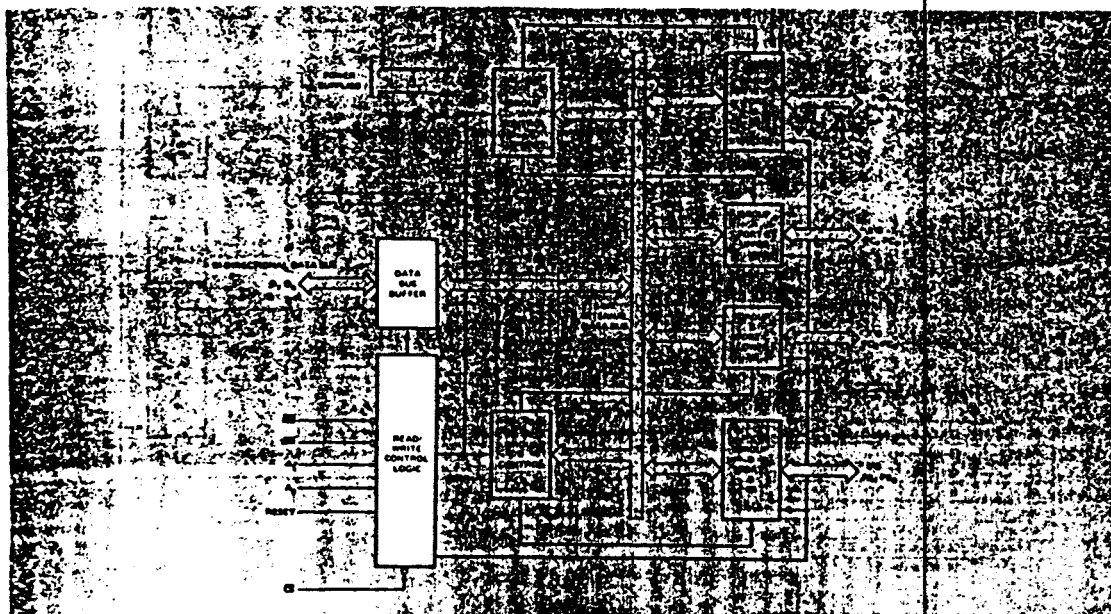
Write: A "low" on this input pin enables the CPU to write Data or Control words into the 8255.

(A_0 and A_1)

Port Select 0 and Port Select 1: These input signals, in conjunction with the \overline{RD} and \overline{WR} inputs, control the selection of one of the three ports or the Control Word Register. They are normally connected to the least significant bits of the Address Bus (A_0 and A_1).

8255 BASIC OPERATION

A_1	A_0	\overline{RD}	\overline{WR}	\overline{CS}	INPUT OPERATION (READ)
0	0	0	1	0	PORT A - DATA BUS
0	1	0	1	0	PORT B - DATA BUS
1	0	0	1	0	PORT C - DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS - PORT A
0	1	1	0	0	DATA BUS - PORT B
1	0	1	0	0	DATA BUS - PORT C
1	1	1	0	0	DATA BUS - CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS - 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS - 3-STATE



8255 Block Diagram

(RESET)

Reset: A "high" on this input clears all internal registers including the Control Register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255. The control word contains information such as "mode", "bit set", "bit reset" etc. that initializes the functional configuration of the 8255.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A — Port A and Port C upper (C7-C4)

Control Group B — Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

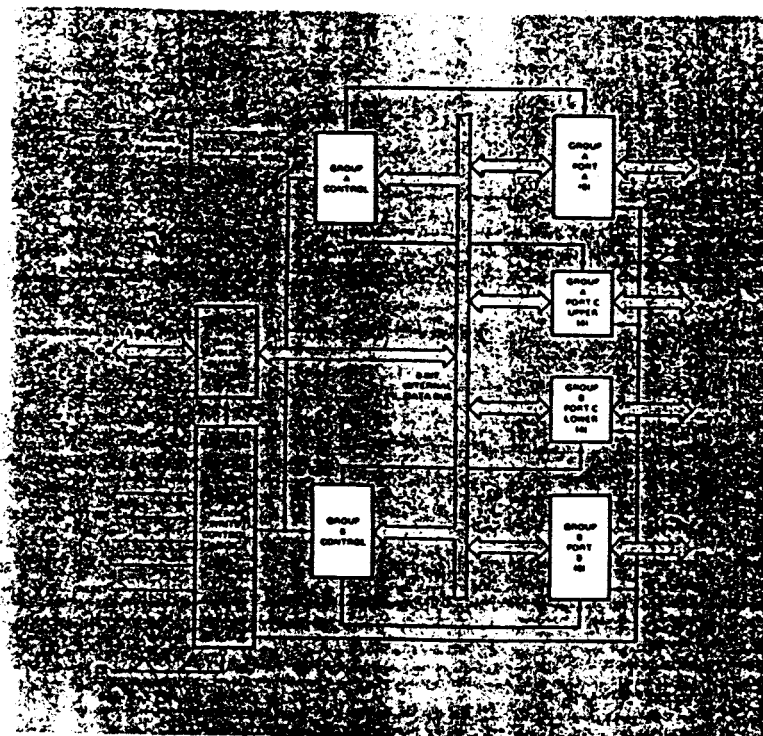
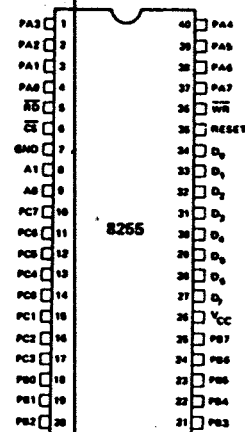
Ports A, B, and C

The 8255 contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B.

8255 BLOCK DIAGRAM**PIN CONFIGURATION****PIN NAMES**

D ₇ -D ₀	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A ₀ , A ₁	PORT ADDRESS
PA ₇ -PA ₀	PORT A (8BIT)
PB ₇ -PB ₀	PORT B (8BIT)
PC ₇ -PC ₀	PORT C (8BIT)
V _{CC}	+5 VOLTS
GND	0 VOLTS

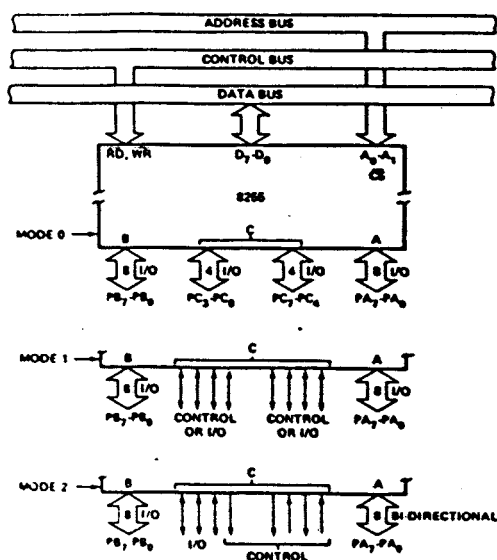
Mode Selection

There are three basic modes of operation that can be selected by the system software:

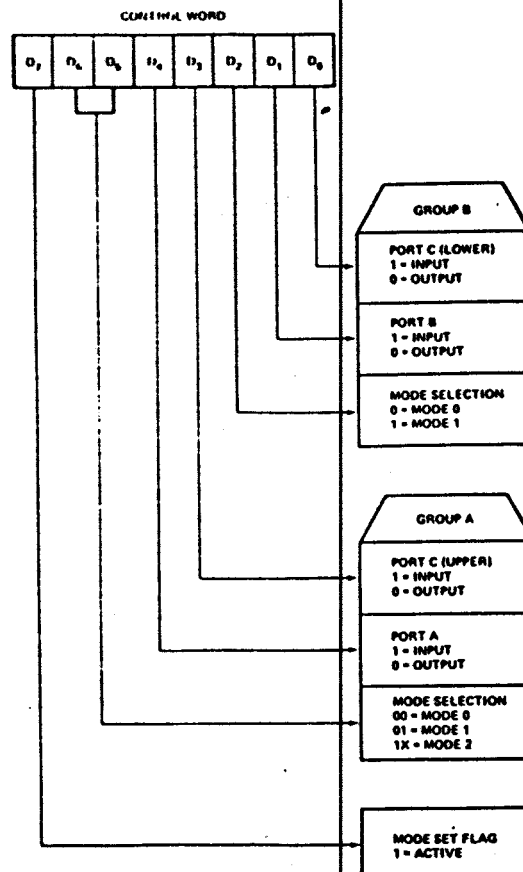
- Mode 0 – Basic Input/Output
- Mode 1 – Strobed Input/Output
- Mode 2 – Bi-Directional Bus

When the RESET input goes "high" all ports will be set to the Input mode (i.e., all 24 lines will be in the high impedance state). After the RESET is removed the 8255 can remain in the Input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single OUTPUT instruction. This allows a single 8255 to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.



Basic Mode Definitions and Bus Interface

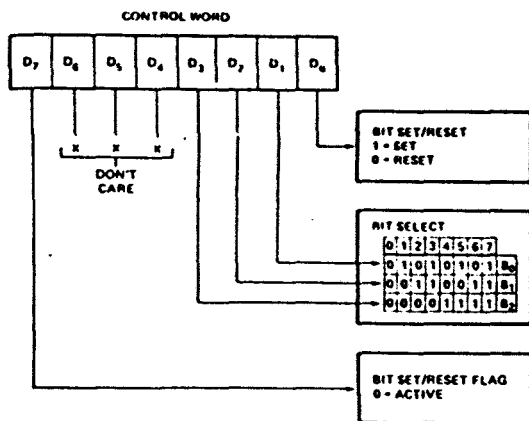


Mode Definition Format

The Mode definitions and possible Mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255 has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTPUT instruction. This feature reduces software requirements in Control-based applications.



When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255 is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit set/reset function of Port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) - INTE is SET - Interrupt enable

(BIT-RESET) - INTE is RESET - Interrupt disable

Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Bit Set/Reset Format

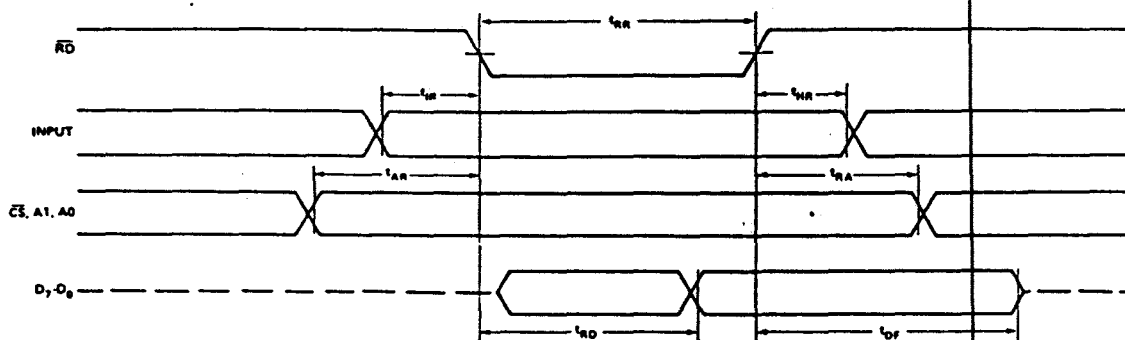
Operating Modes

Mode 0 (Basic Input/Output)

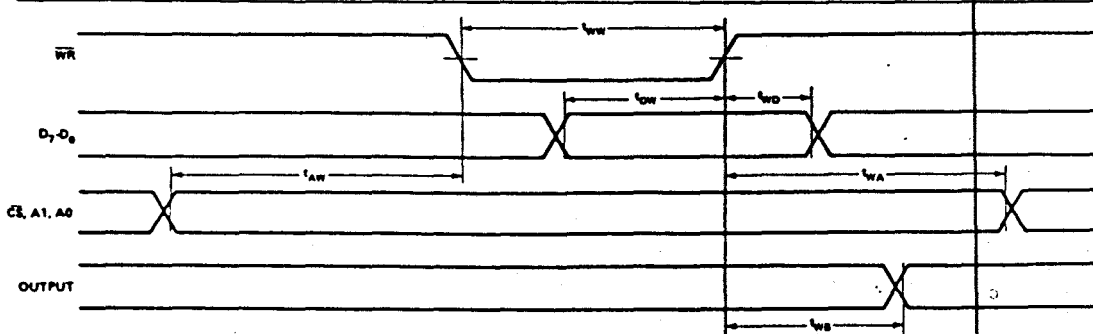
This functional configuration provides simple Input and Output operations for each of the three ports. No "hand-shaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



Mode 0 (Basic Input)



Mode 0 (Basic Output)

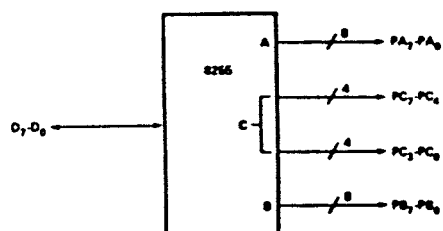
MODE 0 PORT DEFINITION CHART

A		B		GROUP A		GROUP B		
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 CONFIGURATIONS

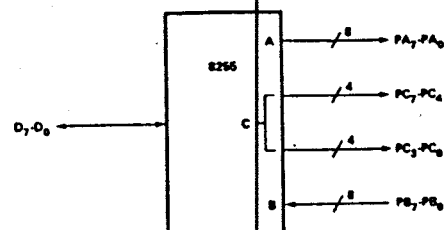
CONTROL WORD #0

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0	0



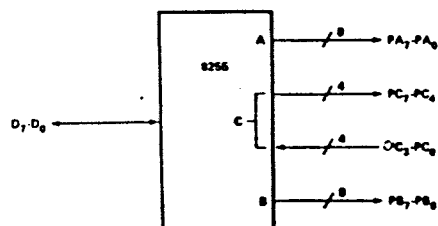
CONTROL WORD #2

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	0



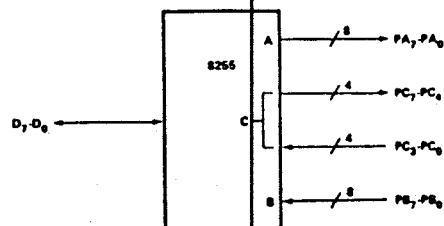
CONTROL WORD #1

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0	1



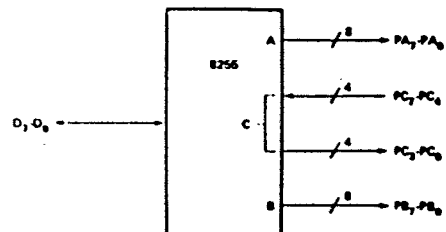
CONTROL WORD #3

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	1



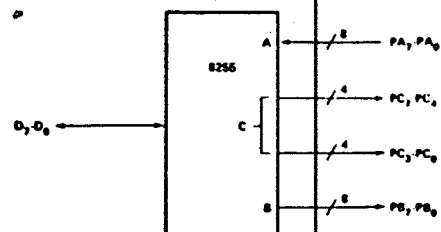
CONTROL WORD #4

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	0



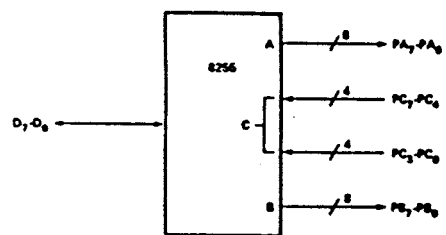
CONTROL WORD #8

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	0



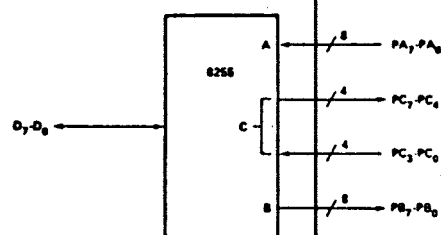
CONTROL WORD #5

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	1



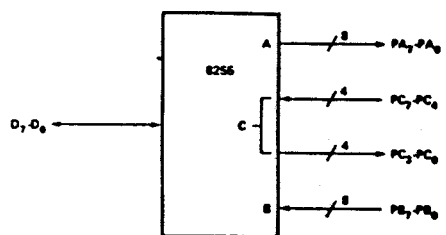
CONTROL WORD #9

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	1



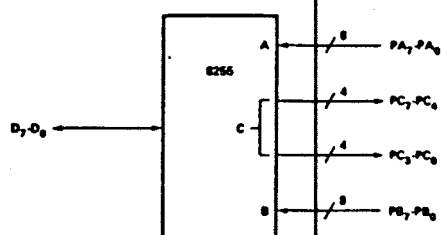
CONTROL WORD #6

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	0



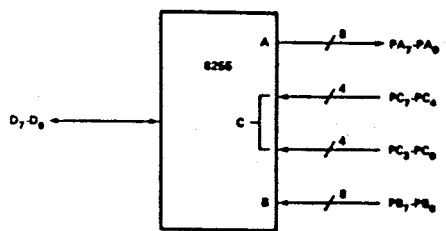
CONTROL WORD #10

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	1	0



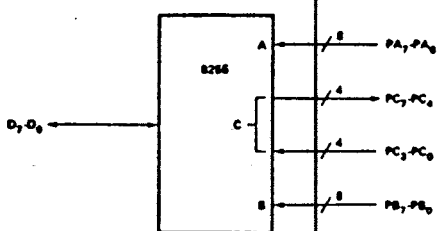
CONTROL WORD #7

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	1



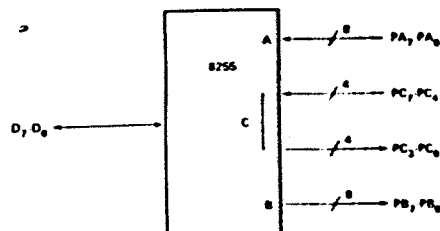
CONTROL WORD #11

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	1	1



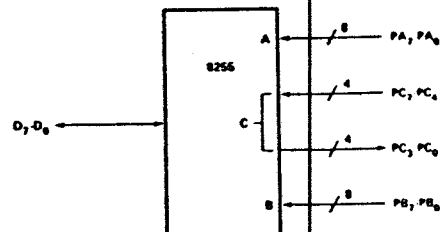
CONTROL WORD #12

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	0	0



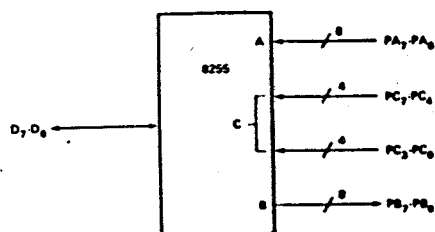
CONTROL WORD #14

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	1	0



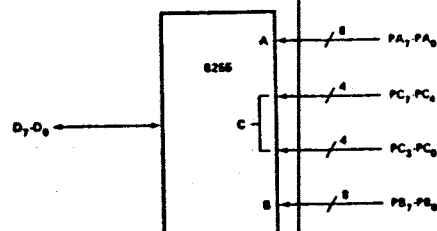
CONTROL WORD #13

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	0	1



CONTROL WORD #15

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	1	0	1	1



Operating Modes

Mode 1 (Strobed Input/Output)

This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

\overline{STB} (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by \overline{STB} input being low and is reset by the rising edge of the \overline{RD} input.

INTR (Interrupt Request)

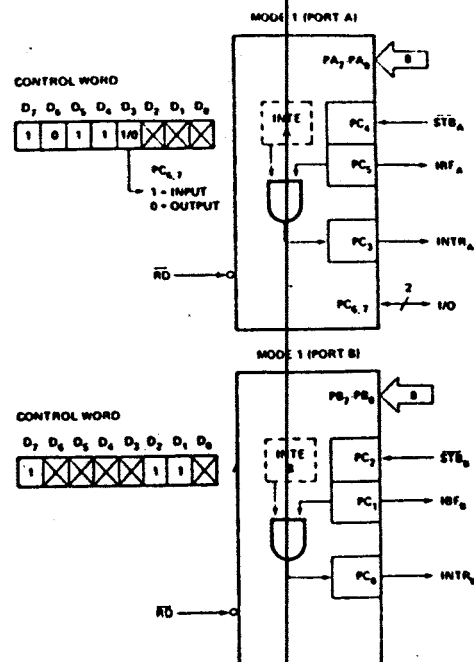
A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the \overline{STB} is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of \overline{RD} . This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

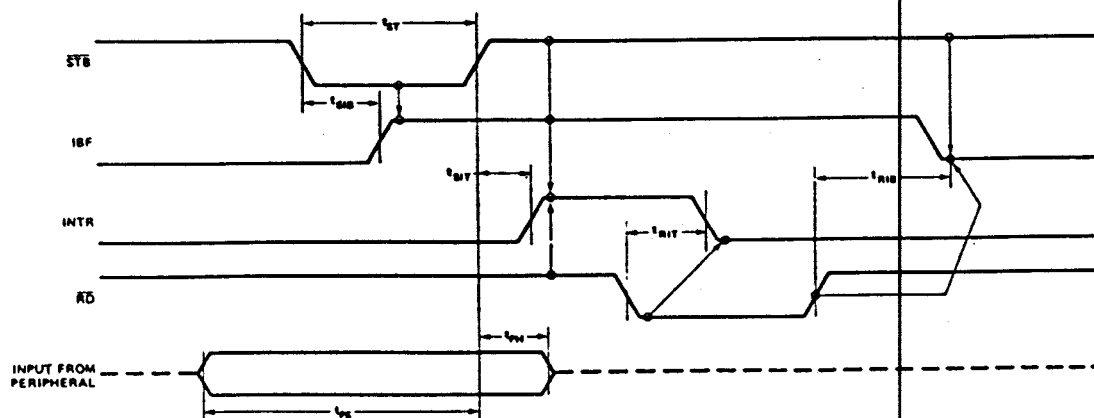
Controlled by bit set/reset of PC_4 .

INTE B

Controlled by bit set/reset of PC_2 .



Mode 1 Input



Mode 1 (Strobed Input)

$\overline{\text{OBF}}$ (Output Buffer Full F/F)

The $\overline{\text{OBF}}$ output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by $\overline{\text{ACK}}$ input being low.

 $\overline{\text{ACK}}$ (Acknowledge Input)

A "low" on this input informs the 8255 that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request)

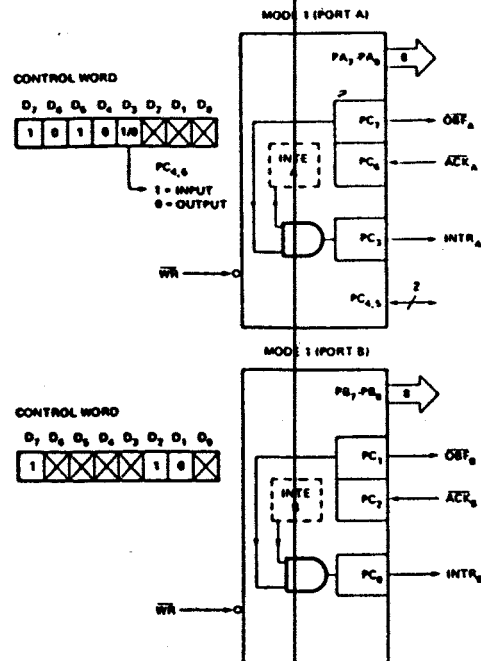
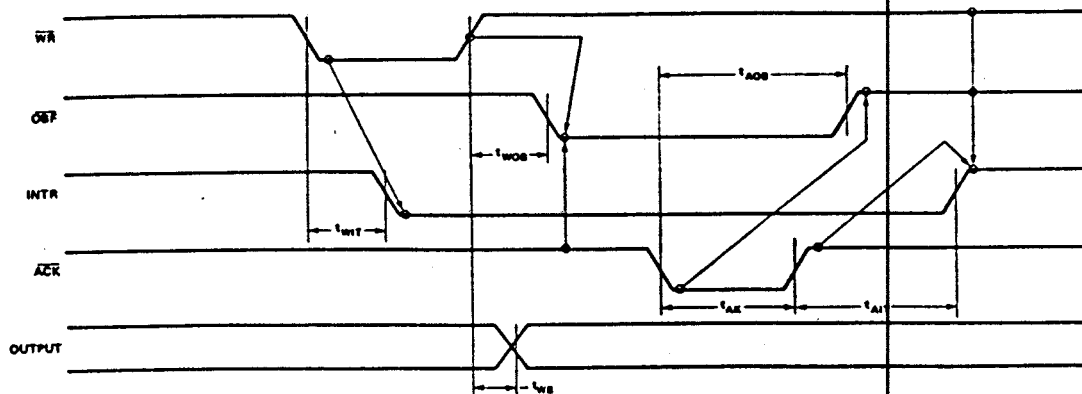
A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set by $\overline{\text{ACK}}$ is a "one", $\overline{\text{OBF}}$ is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC₆.

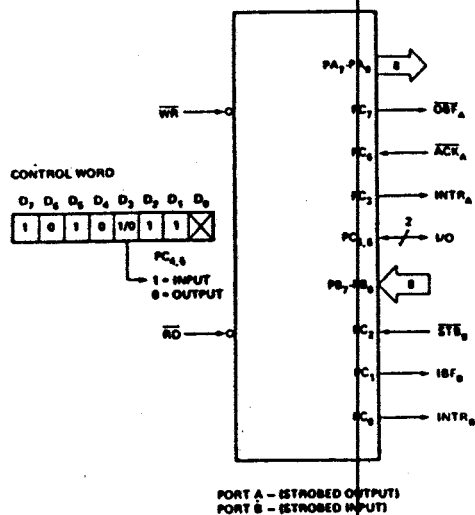
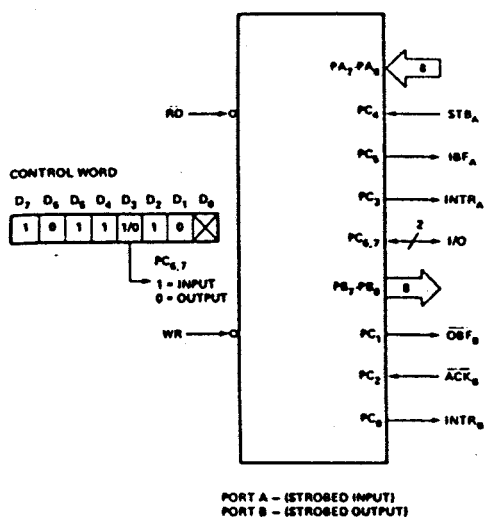
INTE B

Controlled by bit set/reset of PC₂.

**Mode 1 Output****Mode 1 (Strobed Output)**

Combinations of Mode 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.



Operating Modes

Mode 2 (Strobed Bi-Directional Bus I/O)

This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to Mode 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bi-Directional Bus I/O Control Signal Definition

INTR (Interrupt Request)

A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

OBF (Output Buffer Full)

The OBF output will go "low" to indicate that the CPU has written data out to Port A.

ACK (Acknowledge)

A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high-impedance state.

INTE 1 (The INTE Flip-Flop associated with OBF)

Controlled by bit set/reset of PC₆.

Input Operations

STB (Strobe Input)

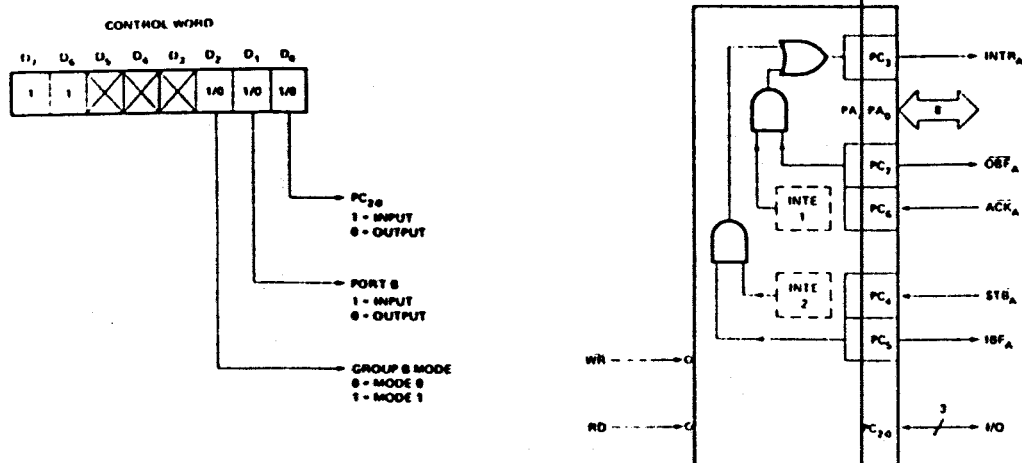
A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that data has been loaded into the input latch.

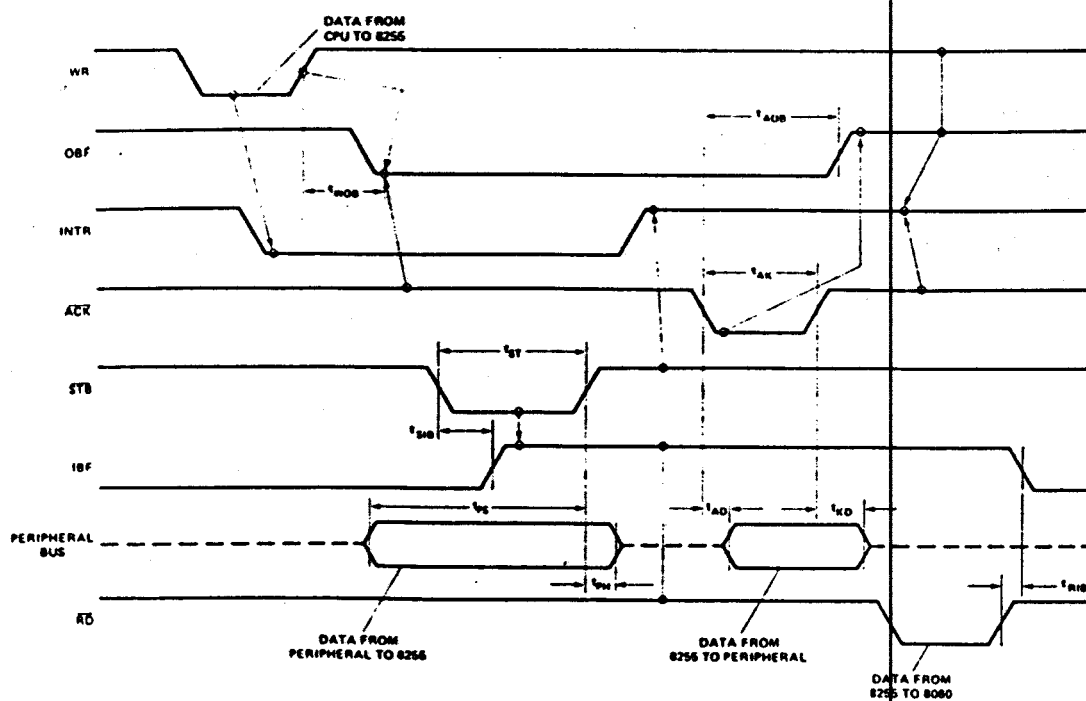
INTE 2 (The INTE Flip-Flop associated with IBF)

Controlled by bit set/reset of PC₄.



Mode 2 Control Word

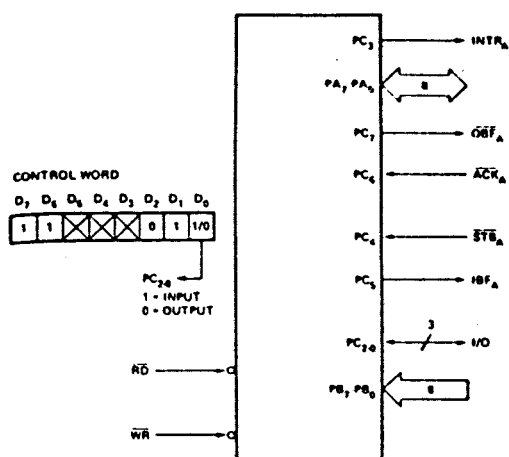
Mode 2



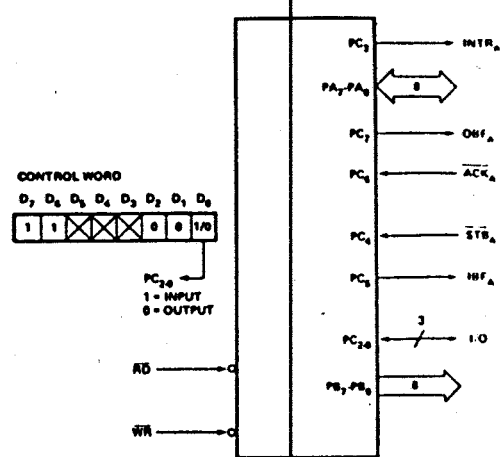
Mode 2 (Bi-directional)

NOTE: Any sequence where \overline{WR} occurs before \overline{ACK} and \overline{STB} occurs before \overline{RD} is permissible.
 $(INTR = IBF \cdot MASK \cdot \overline{STB} \cdot \overline{RD} \cdot \overline{OBF} \cdot MASK \cdot \overline{ACK} \cdot \overline{WR})$

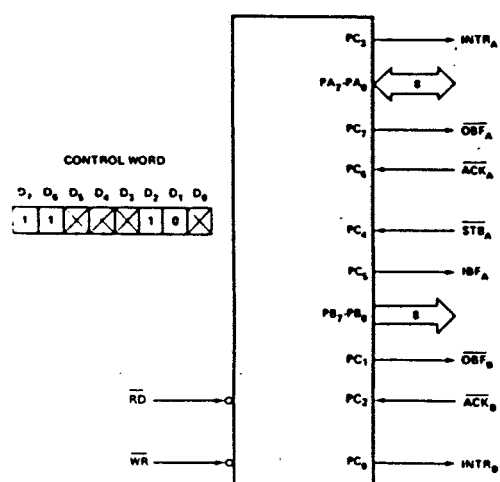
MODE 2 AND MODE 0 (INPUT)



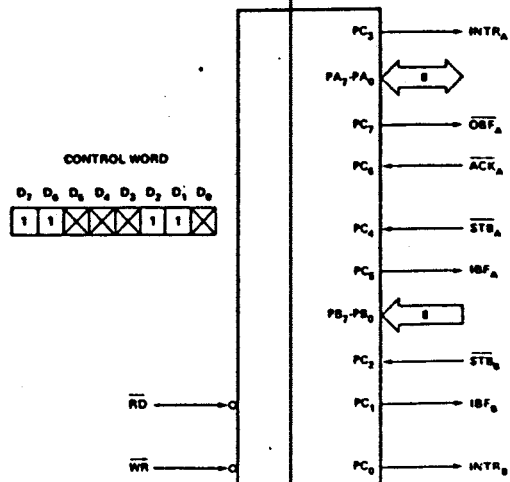
MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)



MODE DEFINITION SUMMARY TABLE

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA ₀	IN	OUT	IN	OUT	←→	
PA ₁	IN	OUT	IN	OUT	←→	
PA ₂	IN	OUT	IN	OUT	←→	
PA ₃	IN	OUT	IN	OUT	←→	
PA ₄	IN	OUT	IN	OUT	←→	
PA ₅	IN	OUT	IN	OUT	←→	
PA ₆	IN	OUT	IN	OUT	←→	
PA ₇	IN	OUT	IN	OUT	←→	
PB ₀	IN	OUT	IN	OUT	—	
PB ₁	IN	OUT	IN	OUT	—	
PB ₂	IN	OUT	IN	OUT	—	
PB ₃	IN	OUT	IN	OUT	—	
PB ₄	IN	OUT	IN	OUT	—	
PB ₅	IN	OUT	IN	OUT	—	
PB ₆	IN	OUT	IN	OUT	—	
PB ₇	IN	OUT	IN	OUT	—	
PC ₀	IN	OUT	INTR _B	INTR _B	I/O	
PC ₁	IN	OUT	IBF _B	OBFB	I/O	
PC ₂	IN	OUT	STB _B	ACK _B	I/O	
PC ₃	IN	OUT	INTR _A	INTR _A	INTR _A	
PC ₄	IN	OUT	STB _A	I/O	STB _A	
PC ₅	IN	OUT	IBF _A	I/O	IBF _A	
PC ₆	IN	OUT	I/O	ACK _A	ACK _A	
PC ₇	IN	OUT	I/O	OBFA	OBFA	

MODE 0
OR MODE 1
ONLY

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs —

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs —

Bits in C upper (PC₇-PC₄) must be individually accessed using the bit set/reset function.

Bits in C lower (PC₃-PC₀) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

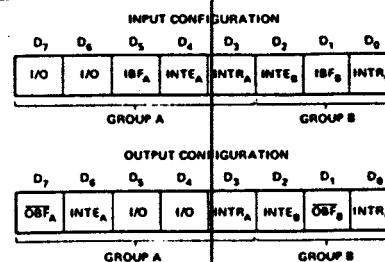
Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

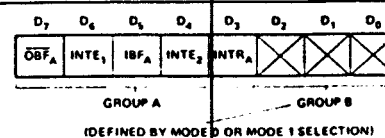
In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C

allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.



Mode 1 Status Word Format

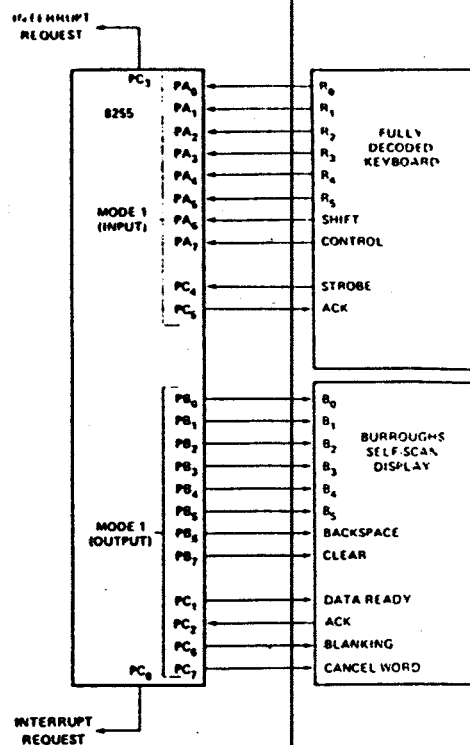
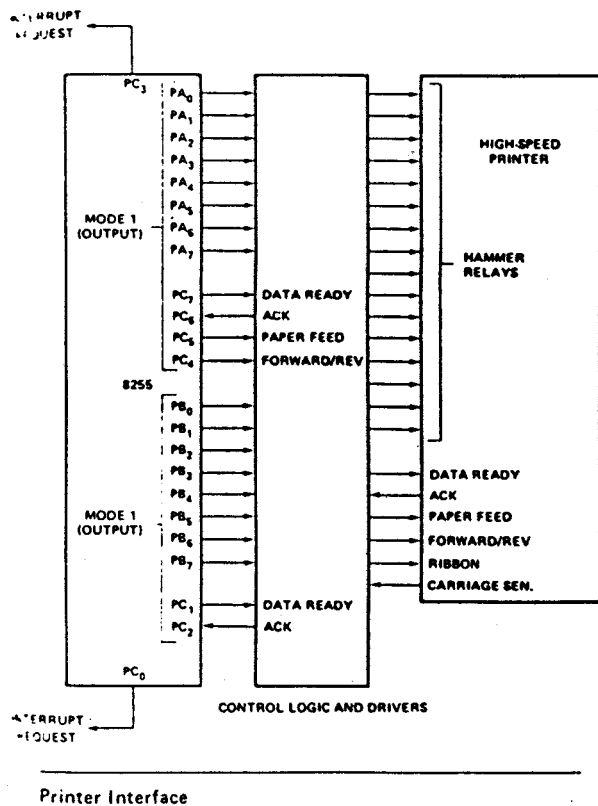


Mode 2 Status Word Format

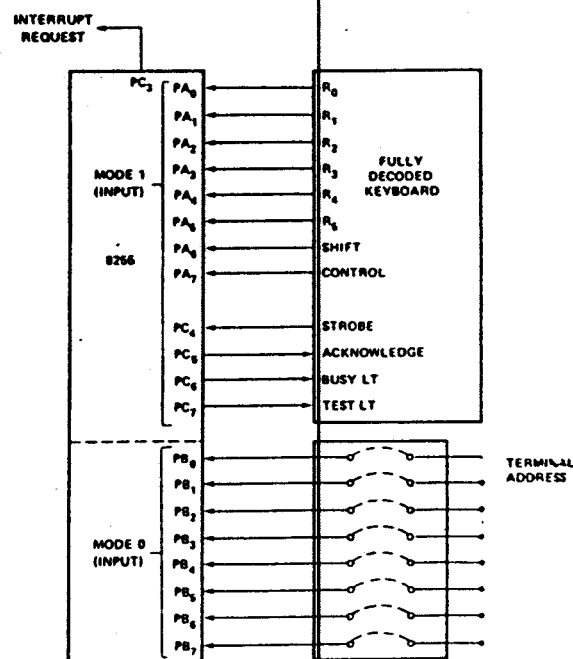
APPLICATIONS OF THE 8255

The 8255 is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

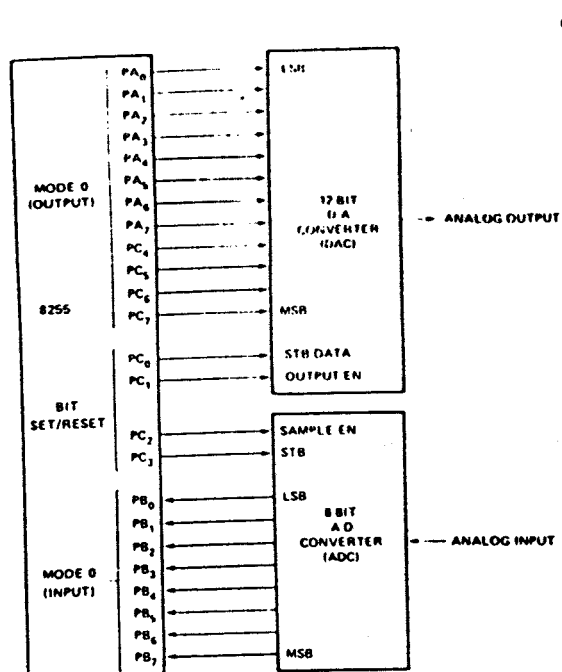
Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255 is programmed by the I/O service routine and becomes an extension of the systems software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the Detailed Operational Description, a control word can easily be developed to initialize the 8255 to exactly "fit" the application. Here are a few examples of typical applications of the 8255.



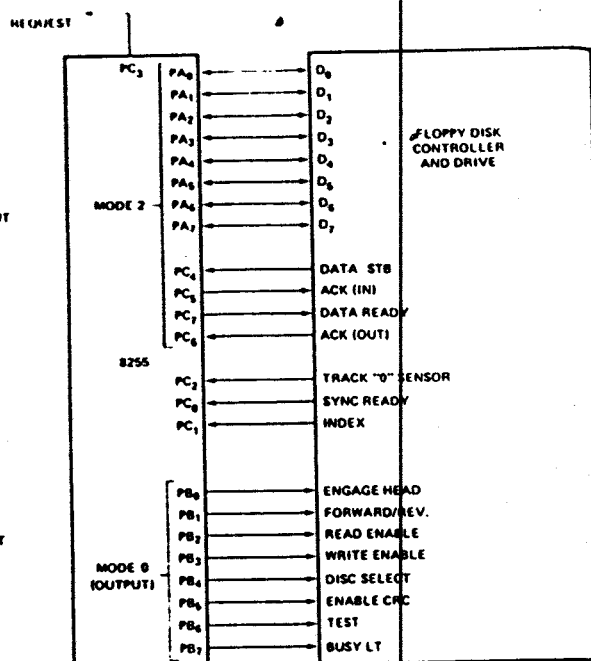
Keyboard and Display Interface



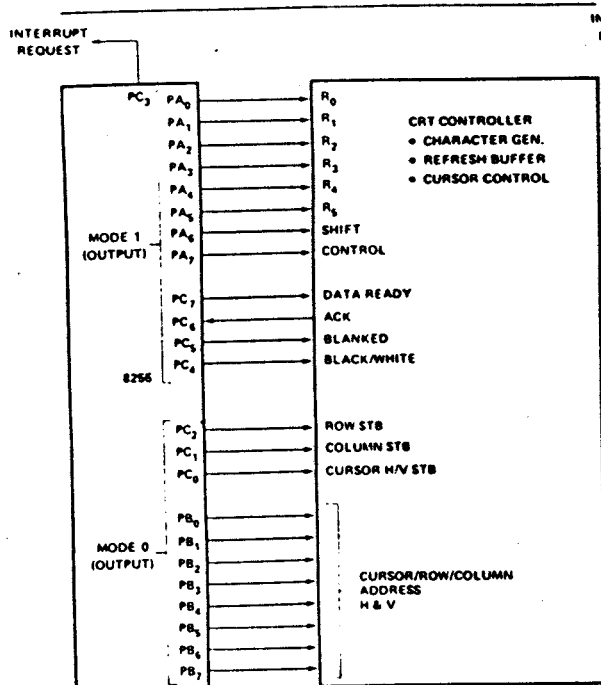
Keyboard and Terminal Address Interface



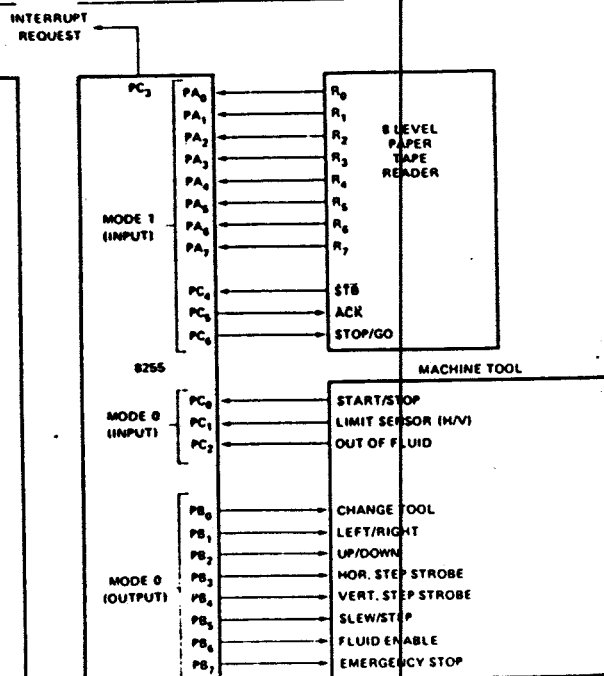
Digital to Analog, Analog to Digital



Basic Floppy Disc Interface



Basic CRT Controller Interface



Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. 0°C to 70°C
 Storage Temperature -65°C to +150°C
 Voltage on Any Pin

With Respect to Ground. -0.5V to +7V
 Power Dissipation 1 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5V \pm 5\%$; GND = 0V

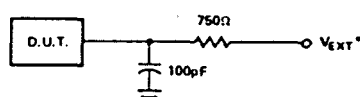
SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
$V_{OL}(DB)$	Output Low Voltage (Data Bus)		0.45	V	$I_{OL} = 2.5\text{mA}$
$V_{OL}(PER)$	Output Low Voltage (Peripheral Port)		0.45	V	$I_{OL} = 1.7\text{mA}$
$V_{OH}(DB)$	Output High Voltage (Data Bus)	2.4		V	$I_{OH} = -400\mu\text{A}$
$V_{OH}(PER)$	Output High Voltage (Peripheral Port)	2.4		V	$I_{OH} = -200\mu\text{A}$
$I_{DAR}^{(1)}$	Darlington Drive Current	-1.0	-4.0	mA	$R_{EXT} = 750\Omega$; $V_{EXT} = 1.5V$
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		± 10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		± 10	μA	$V_{OUT} = V_{CC}$ to 0V

Note 1: Available on any 8 pins from Port B and C.

CAPACITANCE $T_A = 25^\circ\text{C}$; $V_{CC} = \text{GND} = 0V$

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance			10	pF	$f_c = 1\text{MHz}$
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND

TEST LOAD CIRCUIT (FOR DB)



* V_{EXT} IS SET AT VARIOUS VOLTAGES DURING TESTING TO GUARANTEE THE SPECIFICATION.

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C ; $V_{CC} = +5V \pm 5\%$; $GND = 0V$

BUS PARAMETERS:

READ:

NOTE:
The 8255A-5 specifications are not final. Some parametric limits are subject to change.

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{AR}	Address Stable Before READ	0		50		ns
t_{RA}	Address Stable After READ	0		30		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ ⁽¹⁾		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

WRITE:

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{AW}	Address Stable Before WRITE	0		50		ns
t_{WA}	Address Stable After WRITE	20		30		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		250		ns
t_{WD}	Data Valid After WRITE	30		30		ns

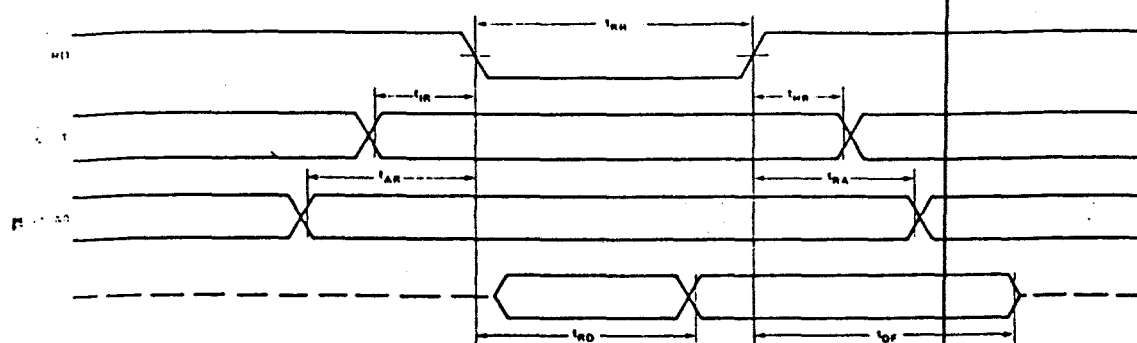
OTHER TIMINGS:

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{WB}	WR = 1 to Output ⁽¹⁾		350		350	ns
t_{IR}	Peripheral Data Before RD	0				ns
t_{IR}	Peripheral Data After RD	0				ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output ⁽¹⁾		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0 ⁽¹⁾		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1 ⁽¹⁾		350		350	ns
t_{SIB}	STB = 0 to IBF = 1 ⁽¹⁾		300		300	ns
t_{RIB}	RD = 1 to IBF = 0 ⁽¹⁾		300		300	ns
t_{RIT}	RD = 0 to INTR = 0 ⁽¹⁾		400		400	ns
t_{SIT}	STB = 1 to INTR = 1 ⁽¹⁾		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1 ⁽¹⁾		350		350	ns
t_{WIT}	WR = 0 to INTR = 0 ⁽¹⁾		850		850	ns

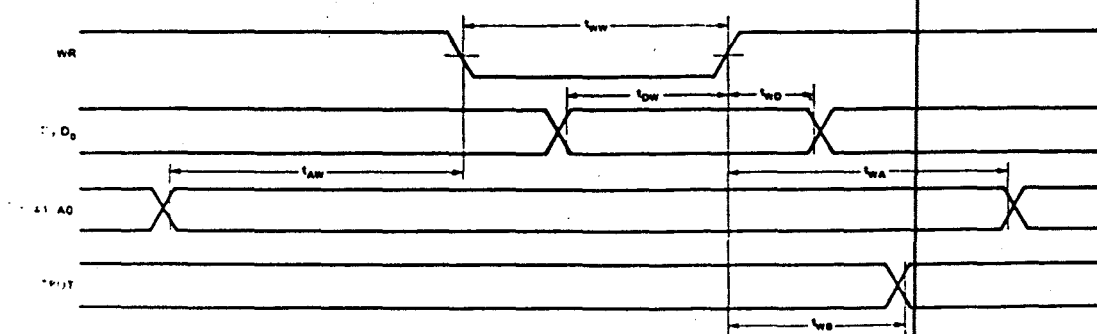
- Notes: 1. Test Conditions: 8255A: $C_L = 100\text{pF}$; 8255A-5: $C_L = 150\text{pF}$.
2. Period of Reset pulse must be at least 50 μs during or after power on. Subsequent Reset pulse can be 500 ns min.



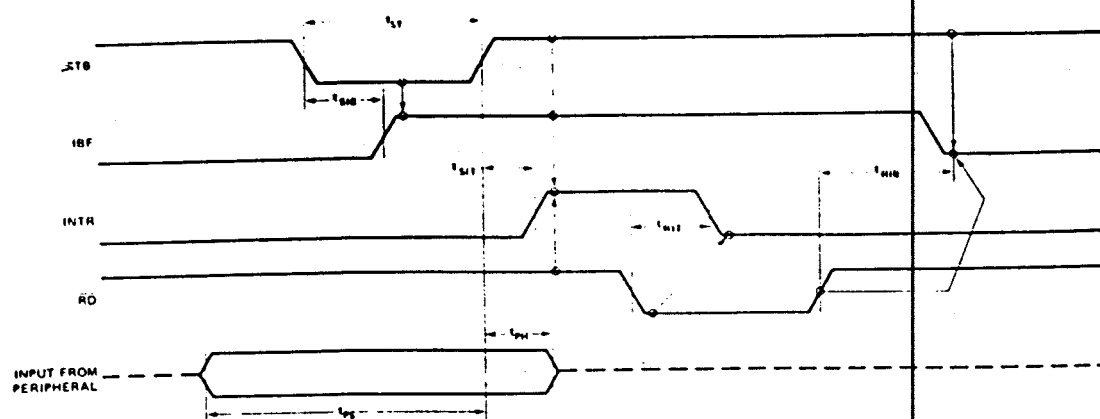
Input Waveforms For A.C. Tests



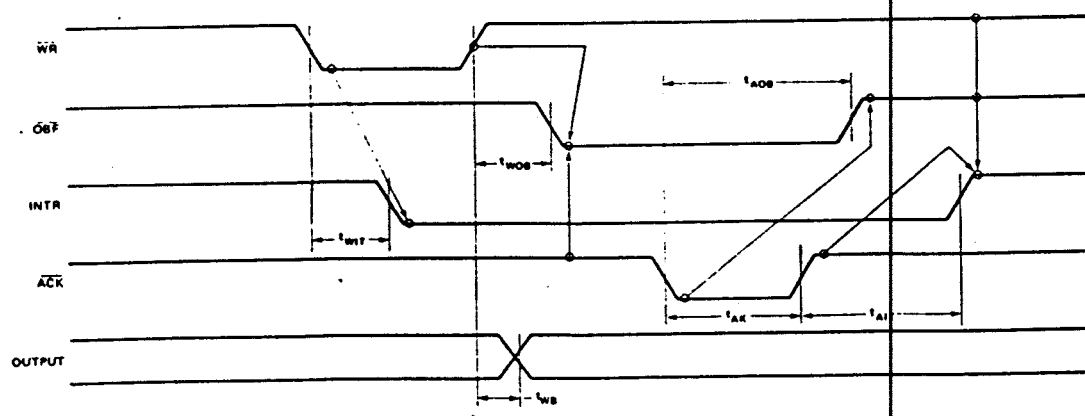
Mode 0 (Basic Input)



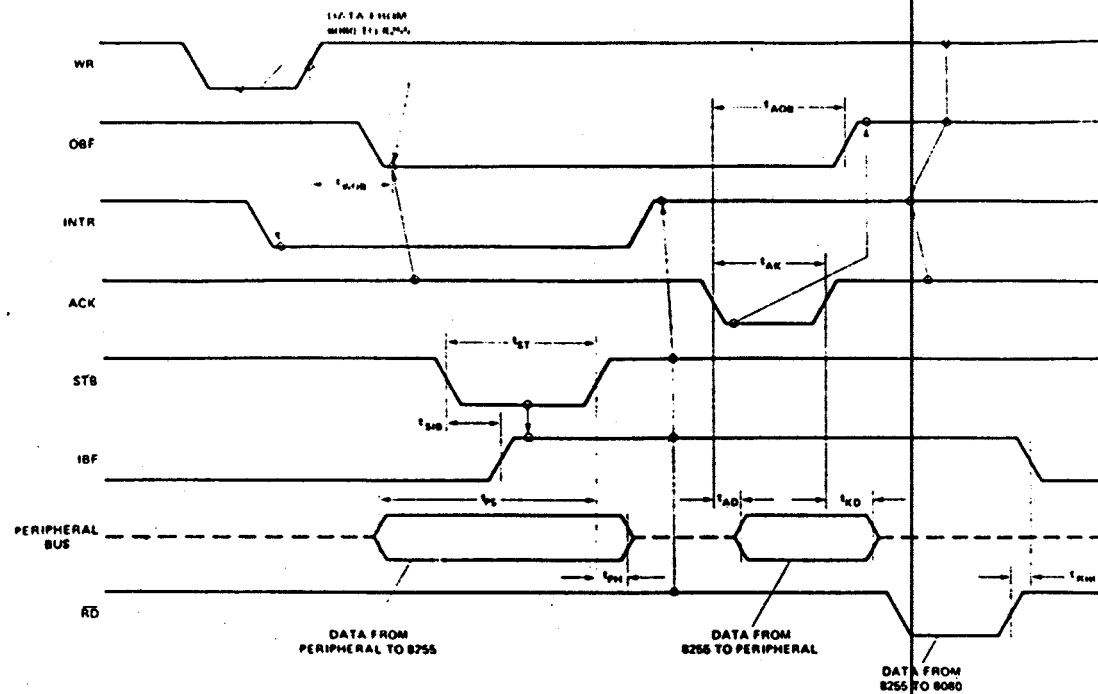
Mode 0 (Basic Output)



Mode 1 (Strobed Input)



Mode 1 (Strobed Output)



Mode 2 (Bi-directional)

NOTE: Any sequence where \overline{WR} occurs before \overline{ACK} and \overline{STB} occurs before \overline{RD} is permissible.
 $(INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + OBF \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR})$